UNIVERSIDAD CARLOS III MADRID

PROYECTO FIN DE CARRERA
INGENIERÍA INDUSTRIAL

"Desarrollo de una interfaz gráfica en MatLab para la aplicación de modelos de Regresión Local Polinómica"

AUTOR: JOSÉ FRANCISCO ESCRIBANO MOLINA

TUTORA: ISABEL GONZÁLEZ FARIAS
Dpto. Ingeniería Mecánica

Octubre 2009

Desarrollo de una interfaz gráfica en Matlab para la aplicación de modelos de Regresión Local Polinómica

José Francisco Escribano Molina
Titulación
Ingeniería Industrial

Fecha: Octubre 2009

Índice

1. Introducción

1.1. Introducción

Para determinar el valor de una magnitud física se realizan medidas de ella de forma directa, o a través de otras magnitudes relacionadas con ella. La realización de estas mediciones trae consigo la obtención de multitud de datos y, por consiguiente una variabilidad en los resultados que hace necesario el uso de técnicas estadísticas para su análisis. En muchas ocasiones, la realización de estas mediciones no es factible debido, por ejemplo a el costo de mismo, y por lo tanto se desea predecir nuevos valores a partir de mediciones ya realizadas. También, en muchas ocasiones, no se conoce la expresión matemática que liga ciertas variables, lo que hace necesario el uso de modelos, estimados a partir de los datos, que permitan definir esas relaciones entre variables.

Dentro de las técnicas estadísticas más utilizadas para encontrar la relación entre variables y predecir nuevos valores, se encuentran los modelos paramétricos, por ejemplo, la regresión lineal univariante o múltiple, la regresión polinómica, etc. Estos modelos se basan en hipótesis que deben cumplir los datos para que el modelo sea adecuado, por ejemplo, hipótesis relacionadas con la distribución de las variables. Esta característica hace que los modelos paramétricos no se puedan aplicar de forma generalizada. Si bien existen muchos fenómenos del mundo real que pueden modelarse de esta manera, para el tratamiento de ciertos problemas estas técnicas paramétricas no son las más adecuadas, debido a la complejidad de las relaciones entre variables.

En los casos en donde los modelos paramétricos no sean adecuados es posible aplicar los modelos no paramétricos. Estos modelos son procedimientos de inferencia estadística que no realizan una suposición con respecto a la forma de la distribución de probabilidad de los datos. Si bien en los modelos no paramétricos también aparecen parámetros, éstos están definidos de manera más general que en el caso de los modelos paramétricos. Los modelos no paramétricos son técnicas para el ajuste de funciones que resultan útiles, por ejemplo, cuando existe poco conocimiento a priori acerca de la forma de dichas funciones. Los fundamentos de estos métodos son antiguos pero sólo lograron el estado actual de desarrollo gracias a los avances de la computación y los estudios por simulación han permitido evaluar sus comportamientos. Entre las técnicas paramétricas más conocidas se encuentran las Redes Neuronales, la Regresión Local Polinómica (RLP), las splines, etc.

En este Proyecto Fin Carrera se ha desarrollado una interfaz gráfica denominada "MathNonParametrics"que permite estimar modelos basadas en Regresión Local Polinómica. La interfaz tiene un manejo sencillo, sin necesidad de que el usuario conozca en detalle toda la teoría relacionada con esta técnica. Además proporcionas las herramientas necesarias para analizar la bondad de ajuste del modelo, mediante resultados numéricos y gráficos; y ayu-

das on-line que puedan guiar al usuario. Para la realización de dicha interfaz se ha usado la herramienta Guide que proporciona el programa de cálculo MatLab.

1.2. Objetivos

EL objetivo principal de este Proyecto Fin Carrera es desarrollar una interfaz gráfica en Matlab, que se ha denominado "MathNonParametrics", que permita encontrar el mejor modelo de Regresión Local Polinómica(RLP) para un conjunto de datos determinado. Asimismo, la interfaz debe permitir predecir nuevos valores, utilizando el modelo óptimo estimado.

La interfaz permitirá a cualquier usuario, incluso aquéllos con pocos conocimientos de RLP, estimar un modelo de RLP de manera rápida y sencilla. Para ello, se brinda al usuario una serie de opciones que le permiten estimar los mejores parámetros del modelo sin necesidad de acceder ni manejar los programas de Matlab que pueden resultar muy complicados para las personas que no tienen mucho conocimiento de programación. La interfaz gráfica incluirá todos los pasos necesarios para ejecutar con éxito el modelo de RLP, desde el análisis previo de los datos hasta la representación gráfica de los resultados una vez ejecutado el modelo.

Además de este objetivo, se quiere desarrollar una interfaz que cumpla con las siguientes características:

1. Sencilla de utilizar.
2. Desarrollo de forma secuencial, de forma que el usuario se sienta guiado a través de la interfaz.
3. Análisis previo de datos.
4. Presentación de resultados numéricos y gráficos para el análisis de la bondad de ajuste del modelo.
5. Opciones de guardar e imprimir cualquier información que se utliice o genere.
6. Ayudas on-line, que permitan guiar al usuario.

1.3. Estructura

En los siguientes capítulos se describirán algunos conceptos que serán de utilidad y se describirán detalladamente las opciones de la interfaz gráfica MathNonParametrics.

En el capítulo 2 se describirán los fundamentos teóricos relacionados con los modelos de Regresión Local Polinómica.

En el capítulo 3 se describirá el uso de la GUI de MatLab, que será utilizada para el desarrollo de la interfaz gráfica.

En el capítulo 4 de presenta la interfaz gráfica creada a partir del programa MatLab denominada *MathNonParametrics*.

En el capítulo 5 se presentan dos aplicaciones de la interfaz MatNonParametrics a datos reales. El primero relacionado con la estimación de la curva de potencia de un aerogenerador, y el segundo relacionado con un proceso de taladrado.

En el capítulo 6 se presentan las conclusiones y las futuras líneas de trabajo.

Por último se incluye la bibliografía consultada.

2. Regresión polinómica local con Kernels

2.1. Introducción

El objetivo de un análisis de regresión es encontrar una función de relación entre dos variables correladas X, variable independiente, e Y, variable dependiente. Dado un conjunto de n observaciones independientes $\{(X_i, Y_i)\}_{i=1}^n$, el modelo de regresión es expresado, en general, como:

$$Y_i = \mu(X_i) + \varepsilon_i = m\left(X_i\right) + \varepsilon_i; \quad i = 1, 2, 3, ..., n \tag{1}$$

donde ε_i es una variable aleatoria que representa el efecto del resto de variables que influyen en Y_i, y que denominaremos término error o ruido. Inicialmente vamos a suponer las siguientes propiedades para este término de error

$$E(\varepsilon_i) = 0, E(\varepsilon_i^2) = \sigma^2, E(\varepsilon_i \varepsilon_j) = 0; \text{ si } i \neq j. \tag{2}$$

El término $m(\bullet)$ es la función de regresión. Esta función evaluada en x es, usando (2) igual a:

$$m(x) = E(Y|X = x). \tag{3}$$

La función $m(x)$ representa la parte determinista de (1) dado x. Esta función es, en general, desconocida por lo que hay que estimarla a partir de una muestra de datos. La estimación de $m(x)$ recibe también el nombre de suavizado o smoothing. Por ejemplo, en la figura (1) se representan un conjunto de $n = 1000$ puntos $\{(X_i, Y_i)\}_{i=1}^n$, obtenidos según el modelo:

$$Y_i = \sin^3(2\pi X_i)^3 + \varepsilon_i$$

donde X_i está uniformemente distribuida en $[0, 1]$ y $\varepsilon_i \sim N(0, 0{,}01)$. La figura (1) muestra también, en linea continua, la función de regresión (exacta) $m = \sin^3(2\pi X_i)^3$. Puede observarse que en el intervalo $[0, 0{,}4]$ no existe una dependencia con X, mientras que para $X > 0,4$ la función tiene un pico en $0,63$ y un valle en $0,9$.

El suavizado se puede hacer básicamente de dos maneras. La más utilizada es aplicar un enfoque *paramétrico*, que consiste en asumir que la función $m(\bullet)$ tiene predefinida una forma, por ejemplo, una recta, un polinomio, etc. Un ejemplo típico de un modelo paramétrico es la ecuación de regresión polinómica donde los parámetros son los coeficientes de las variables independientes. Otra alternativa consiste en estimar $m(\bullet)$ de forma *no paramétrica*, es decir, sin asumir ningún modelo a priori. Por consiguiente, un modelo paramétrico puede ser muy útil si se conoce de antemano la manera en la que deben estar relacionados los datos, mientras que el enfoque no paramétrico se podría utilizar para los casos en los que no se conozca de antemano la forma en la que los datos van a estar relacionados.

Un ejemplo de estos enfoques diferentes se muestra en figura (2), donde se representan dos curvas correspondientes a X =demanda del mercado de patatas e Y =ingresos netos. La línea

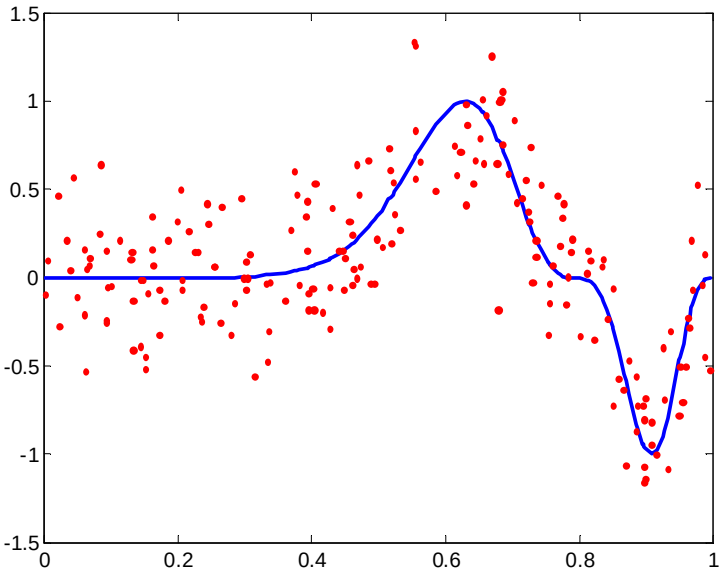


Figura 1: Diagrama de dispersión de datos simulados (puntos) y función de regresión real m(.) (línea continua).

recta corresponde a un modelo paramétrico, mientras que la línea curva ha sido obtenida mediante un modelo no paramétrico de suavizado. Ambos modelos han sido obtenidos a partir de una nube de puntos $X - Y$. En este caso, el modelo lineal no es capaz de modelar una disminución en la demanda de patatas cuando aumentan los ingresos. El modelo no paramétrico sugiere, sin embargo, una relación en forma de U. Luego, los modelos no paramétricos de suavizado proporcionan un método versátil para explorar la posible relación entre variables. Además, permiten realizar predicciones de Y sin necesidad de tener un modelo paramétrico.

En modelos de suavizado es muy importante el tratamiento inicial de los datos. En el caso extremo, si existen datos erróneos pueden llegar a influir de tal forma que hagan que la estructura principal de los datos se vuelva invisible. Se trata de un diagnóstico inicial que permita desvelar los datos no admisibles.

La expresión (3) puede ser explicada también del siguiente modo. Sean X e Y dos variables aleatorias con función de densidad, $f(x)$ y $f(y)$, respectivamente; y con función de densidad conjunta $f(x, y)$. Luego, el modelo (3) es igual a:

$$m(x) = E(Y|X = x) = \int yf(y/x)dy = \int y\frac{f(x, y)}{f(x)}dy. \tag{4}$$

Para poder evaluar la expresión (4) sería necesario conocer, por tanto, las funciones de densidad $f(x)$ y $f(x, y)$.

En el caso de modelos no parámetricos de suavizado estas funciones de densidad no son conocidas, sino que son aproximadas a partir de los datos. Por ejemplo, al observar la

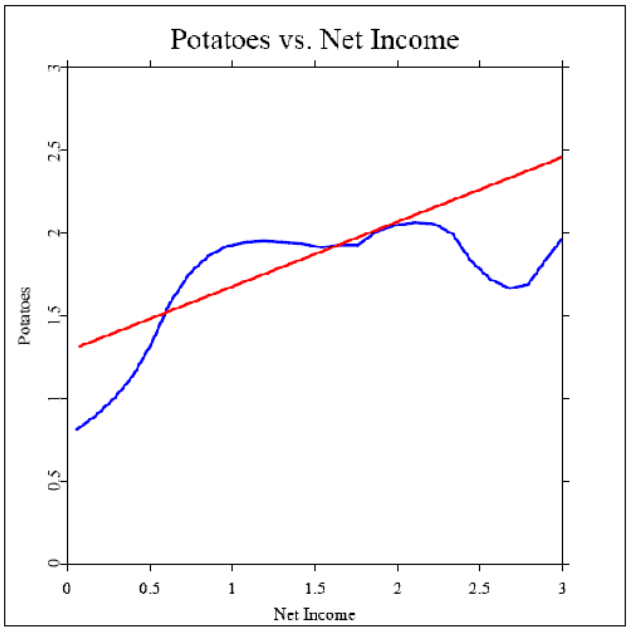


Figura 2: Diferencias de ajuste paramétrico y no paramétrico

figura (1), parece razonable aproximar la curva $m(x)$ en (3) utilizando los puntos cercanos al punto x. Es decir calcular $m(x)$ como el valor medio de la variable respuesta Y calculada con los valores correspondientes a puntos próximos a x. Este "promedio local"debería ser obtenido de forma que las observaciones más próximas a x tuviesen mayor peso que aquéllas observaciones más alejadas de x. Esta idea de promedio local puede ser considerada como la idea fundamental del suavizado no paramétrico y, puede ser expresada como:

$$\hat{m}(x) = \frac{1}{n}\sum_{i=1}^{n} W_i(x)Y_i \tag{5}$$

donde $\{W_i(x)\}_{i=1}^{n}$ es la secuencia de pesos asociados a las observaciones $\{X_i\}_{i=1}^{n}$. A partir de las expresiones (4) y (5), podría concluirse que los pesos W_i pueden ser considerados como una estimación de las funciones de densidad $f(x,y)/f(x)$. Existen distintos métodos para calcular estos pesos W_i, entre ellos por ejemplo, la regresión local ponderada "lowess", método de los K-vecinos, splines, estimador de Nadaraya-Watson basado en funciones Kernel, etc. En las siguientes secciones se describirá el uso de las funciones Kernels en la regresión no paramétrica. Antes, dada la relación entre los pesos W_i y las funciones de densidad $f(x)$ y $f(x,y)$, se describirá en la siguiente sección, la estimación de densidades mediante funciones Kernel .

2.2. Estimación de densidades usando funciones Kernel

2.2.1. Densidades univariantes

Como se ha explicado en el apartado anterior, para definir el modelo no paramétrico en (5) es necesario calcular los pesos W_i que representan una estimación de $f(x,y)/f(x)$. Antes de definir el procedimiento para calcular W_i, en esta sección se explicará cómo estimar una función de densidad $f(x)$ utilizando funciones Kernel. Una forma sencilla de evaluar una función de densidad de forma no paramétrica, podría basarse, por ejemplo, en el histograma. Partiendo de la idea del histograma se podrá intuir de forma mucho más fácil la esencia del método de estimación de densidades basado en funciones Kernel. De forma general, el método del histograma consiste en recoger los datos que están contenidos en intervalos B_j de longitud h, donde J es el número total de intervalos del histograma. De esta forma, se obtiene la frecuencia relativa de cada intervalo. Un histograma puede ser construido a partir de los siguientes pasos:

1. A partir de un valor inicial x_0 construir los J intervalos $B_j = [x_0 + (j-1)h; x_0 + jh]$
2. Contar cuántas observaciones están contenidas en el B_j y denotarlas como n_j.
3. Calcular, para cada intervalo B_j, la densidad $\hat{f}_j = n_j/(nh)$, donde n es el total de observaciones y h es denominado ancho de banda. No confundir esta densidad con la frecuencia relativa n_j/n. La densidad $\hat{f}_j$ representa la proporción de individuos que hay por unidad de medida.
4. Para cada intervalo, dibujamos un rectángulo para cada intervalo de altura $\hat{f}_j$. El conjunto de estos rectángulos es el histograma. Este histograma así construido tiene área unidad, lo que resulta conveniente para introducir la noción de estimación de densidades. (Otra forma de construir histogramas es usando como altura de cada rectángulo la frecuencia relativa n_j/n, o incluso la frecuencia absoluta n_j.)

Formalmente se puede escribir la definición de histograma, para cada punto x, de la siguiente manera:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} \sum_{j} I(X_i \in B_j) I(x \in B_j) \tag{6}$$

donde $I(\cdot)$ es la función indicador, tal que

$$I\ (X_i \in B_j) = \left\{ \begin{array}{l} 1,\ si\ \ X_i \in B_j \\ 0,\ si\ X_i \notin B_j \end{array} \right\}.$$

Como se observa, el parámetro h, denominado ancho de banda (bandwith), es determinante a la hora de crear el histograma y tendrá también repercusión en los estimadores Kernel, como se verá más adelante.

Partiendo de la idea con la que se construye el histograma, la función de densidad puede ser estimada como:

$$f(x) = \frac{1}{n \times (\text{longitud intervalo})} \times \Psi \tag{7}$$

siendo Ψ es el número de observaciones dentro de un intervalo alrededor de x. Este intervalo puede ser definido como $[x - h/2, x + h/2]$, de forma que:

$$\hat{f}_h(x) = \frac{1}{nh} \Psi \left\{ X_i \in [x - h/2, x + h/2] \right\} \tag{8}$$

La función Ψ en (8) puede ser una función que asigne un peso a cada valor X_i en función de su distancia a x. Una función de este tipo es la función Kernel denotada como $K(u)$, donde $u = (x - X_i)/h$. Utilizando el Kernel, la expresión (8) puede ser rescrita como:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h} \right) \tag{9}$$

Es habitual denotar de forma simplificada la función Kernel con respecto al parámetro h como:

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h} \right).$$

Luego, $\hat{f}_h(x) = n^{-1} \sum_{i=1}^{n} K_h(x - X_i)$. Las funciones Kernel satisfacen los siguientes puntos:

- Son simétricas alrededor del 0.
- Cumplen $\int K(x)dx = 1$
- Son funciones de densidad, por tanto la estimación $\hat{f}_h(x)$ es también una función de densidad.
- Si $K(x)$ es m veces diferenciable, también lo es $\hat{f}_h(x)$.
- En general las funciones $K(x)$ son positivas. Pueden considerarse Kernel negativos pero eso implicaría que $\hat{f}_h(x)$. podría ser negativa.
- Si h y la función Kernel son fijadas, la estimación $\hat{f}_h(x)$ para un conjunto de datos es única.

Una de las funciones Kernel más sencilla es el Kernel uniforme definido como:

$$K(u) = \frac{1}{2} I(u), \quad \left\{ \begin{array}{l} I(u) = 1 \text{ si } |u| \leq 1) \\ I(u) = 0 \text{ si } |u| > 1) \end{array} \right\} \tag{10}$$

La función (10) asigna un peso de 1/2 a cada observación X_i cuya distancia a x (punto al cual se desea estimar la función de densidad) no es mayor que h. De esta forma, los puntos que estén alejados de x obtendrán un peso de cero. El Kernel de la expresión (10), asigna a cada observación X_i que verifica $|(x - X_i)/h| \leq 1$, un valor igual a 1/2. Es decir, todos los puntos X_i tienen igual ponderación (un factor constante de 1/2), sin importar lo cerca que estén del valor x. Sin embargo, parece más razonable asignar un peso mayor a los puntos X_i que se encuentren más cerca de x, teniendo en cuenta, por ejemplo la distancia $u = (x - X_i)/h$. Algunos de los Kernel más utilizados y que tienen en cuenta esta distancia u son los siguientes:

- Triangular: $K(u) = (1 - |u|)I(u)$
- Epanechnikov: $K(u) = \frac{3}{4}(1 - u^2)I(u)$
- Quartic: $K(u) = \frac{15}{16}(1 - u^2)^2 I(u)$
- Coseno: $K(u) = \frac{\pi}{4}\cos(\frac{\pi}{2}u)I(u)$
- Gausiano: $K(u) = \frac{1}{2\pi}\exp(-\frac{1}{2}u^2)$.

En la figura (3.a) se representa el histograma de un conjunto de datos X junto con tres estimaciones distintas de la función de densidad. Estas estimaciones se han obtenido aplicando (9) con tres funciones Kernel: Gaussiano, Epanechnikov y Uniforme; todos ellos con un ancho de banda $h = 0,33$. En este caso los Kernel Gaussiano y Epanechnikov producen ajustes muy similares, mientras que el Kernel Uniforme tiene peor comportamiento. En la figura (3.b) se representa la estimación de la función de densidad utilizando el Kernel Gaussiano con dos anchos de banda, $h = 0,15$ y $h = 0,01$. Se puede ver que el parámetro h es un parámetro de suavizado del estimador de la función de densidad. Un valor muy pequeño de h (por ejemplo $h = 0,01$) significa una curva poco suave (overfitting). Por el contrario un h demasiado grande implica una curva muy suave (underfittig). La selección del parámetro h es muy importante en la estimación de la función de densidad. En el siguiente apartado se describirán algunas propiedades estadísticas del estimador $\hat{f}_h(x)$, en (9), que están relacionadas con el parámetro h.

2.2.2. Propiedades estadísticas del estimador $\hat{f}_h(x)$

1. *Sesgo*

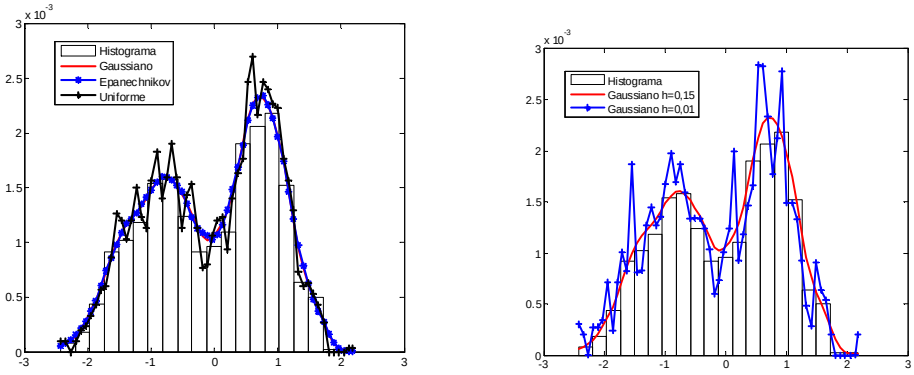


Figura 3: (a)Estimación de la función de densidad utilizando diferentes Kernel con h = 0,33. (b) Estimación de la función de densidad usando Kernel Gaussiano h=0,15 y h=0,01.

Haciendo uso de la definición de sesgo de un estimador, se obtiene la siguiente expresión:

$$\begin{aligned} \text{Sesgo}\left\{\hat{f}_h(x)\right\} &= E\left\{\hat{f}_h(x)\right\} - f(x) \\ &= \frac{1}{n}\sum_{i=1}^{n} E\left\{K_h(x - X_i)\right\} - f(x) \\ &= E\left\{K_h(x - X_i)\right\} - f(x) \end{aligned} \tag{11}$$

A partir de la expresión (11), se obtiene:

$$\text{Sesgo}\left\{\hat{f}_h(x)\right\} = \int \frac{1}{h} K(\frac{x-u}{h}) f(u) du - f(x) \tag{12}$$

Usando la variable $s = (u - x)/h$, la propiedad de simetría de los estimadores Kernel, es decir, $K(-s) = K(s)$, y el desarrollo de Taylor de segundo orden para $f(u)$ alrededor de x se obtiene:

$$\text{Sesgo}\left\{\hat{f}_h(x)\right\} = \frac{h^2}{2} f''(x)\mu_2(K) + o(h^2) \tag{13}$$

donde

$$\mu_2(K) = \int s^2 K(s) ds \tag{14}$$

A partir de (13) se concluye que el sesgo del estimador $\hat{f}_h(x)$ es proporcional al cuadrado del ancho de banda, h^2. Por tanto, a menor h, menor sesgo. Sin embargo, el sesgo de $\hat{f}_h(x)$ es también directamente proporcional al valor de $f''(x)$, que representa la curvatura de la función de densidad en el punto x. En la figura (4) se representa la función de densidad $f(x)$ con línea gruesa y su estimación $\hat{f}_h(x)$ con línea fina. El sesgo es la diferencia entre ambas curvas. Se observa que a mayor curvatura de la función de densidad, mayor es el sesgo. Además, en los valles de la función de densidad se tiene $f'' \succ 0$ alrededor del mínimo local de f, por lo tanto, el sesgo es positivo (la línea

fina está por encima de la línea gruesa). En los picos de la función se tiene el efecto contrario.

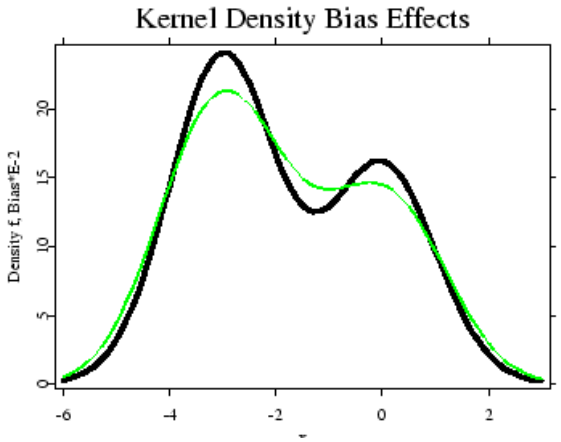


Figura 4: Función de densidad y estimación basado en funciones kernel

2. *Varianza*

La varianza de $\hat{f}_h(x$ es igual a:

$$\begin{aligned} \text{Var}\left\{\hat{f}_h(x)\right\} &= \text{Var}\left\{\frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)\right\} \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\text{Var}\left\{K_h(x - X_i)\right\} \\ &= \frac{1}{n}\text{Var}\left\{K_h(x - X_i)\right\} \\ &= \frac{1}{n}(E\left\{K_h^2(x - X_i)\right\} - [E\left\{K_h(x - X_i)\right\}]^2) \end{aligned} \tag{15}$$

El primer término de esta expresión es igual a:

$$\begin{aligned} \frac{1}{n}(E\left\{K_h^2(x - X_i)\right\} &= \frac{1}{n}\frac{1}{h^2}\int K^2(\frac{x-u}{h})f(u)du \\ E\left\{K_h(x - X_i)\right\} &= f(x) + o(h) \end{aligned} \tag{16}$$

Utlizando el desarrollo de Taylor de segundo orden para $f(u)$ alrededor de x se obtiene:

$$\text{Var}\left\{\hat{f}_h(x)\right\} = \frac{1}{nh}\left\|K\right\|_2^2 f(x) + o(\frac{1}{nh}), \text{ cuando } nh \to \infty \tag{17}$$

donde $\|K\|_2^2 = \int K^2(s)ds$ y representa la norma cuadrática L_2. La varianza de $\hat{f}_h(x)$ es proporcional a h^{-1}. Es decir, para reducir la varianza se tiene que elegir un parámetro h grande.

Luego, la elección del parámetro h implica un compromiso entre sesgo y varianza. A menor h, menor sesgo pero mayor varianza y, a mayor h, menor varianza pero mayor sesgo. Es entonces necesario definir otro parámetro que permita tener en cuenta este efecto.

3. *Error cuadrático medio (MSE-mean square error)*

   El MSE combina el sesgo y la varianza del estimador. Se define como:

$$\begin{aligned} \text{MSE}\left\{\hat{f}_h(x)\right\} &= E\left[\left\{\hat{f}_h(x) - f(x)\right\}^2\right] \\ \text{MSE}\left\{\hat{f}_h(x)\right\} &= \left[\text{Sesgo}\left\{\hat{f}_h(x)\right\}\right]^2 + \text{Var}\left\{\hat{f}_h(x)\right\} \end{aligned}$$

   Utilizando las expresiones (13) y (17),

$$\text{MSE}\left\{\hat{f}_h(x)\right\} = \frac{h^4}{2} f''(x)^2 \mu_2(K)^2 + \frac{1}{nh} \|K\|_2^2 f(x) + o(h^4) + o(\frac{1}{nh}) \tag{18}$$

   A partir de la expresión (18) se puede concluir que el MSE de $\hat{f}_h(x)$ se aproxima a cero para $h \to 0$ y $nh \to \infty$. El valor h óptimo corresponde al mínimo valor de MSE. Si se deriva MSE con respecto a h se descubrirá que las funciones $f(x)$y $f''(x)$ no se eliminan en el proceso de derivación. Estas funciones f y f'', son desconocidas en la práctica, por lo tanto, obtener el valor óptimo de h no es aplicable en la práctica a menos que se encuentre alguna forma de estimar dichas funciones.

4. *Error cuadrático medio integrado (MISE (mean integrated square error)*

$$\begin{aligned} \text{MISE}\left\{\hat{f}_h(x)\right\} &= \int MSE\left\{\hat{f}_h(x)\right\} dx \qquad (19) \\ &= \frac{1}{nh} \|K\|_2^2 + \frac{h^4}{4} \left\{\mu_2(K)\right\}^2 \left\|f''\right\|_2^2 + o(h^4) + o(\frac{1}{nh}) \end{aligned}$$

   Ignorando los términos de mayor grado, el MISE se puede aproximar por wl AMISE:

$$\text{AMISE}\left\{\hat{f}_h(x)\right\} = \frac{1}{nh} \|K\|_2^2 + \frac{h^4}{4} \left\{\mu_2(K)\right\}^2 \left\|f''\right\|_2^2 \tag{20}$$

   Derivando la expresión (20) con respecto a h se tiene el valor óptimo de $h_{ópt}$ que minimiza el AMISE:

$$h_{ópt} = \left(\frac{\|K\|_2^2}{\left\|f''\right\|_2^2 \left\{\mu_2(K)\right\}^2 n}\right)^{1/5} \sim n^{1/5}$$

   Como se puede ver, el valor de $h_{ópt}$ depende de valores desconocidos como f''.

2.2.3. Selección del parámetro de suavizado h

Una forma de seleccionar el parámetro h es minimizando los parámetros descritos en la sección anterior. Los dos procedimientos usados con más frecuencia para encontrar el h óptimo son: el método Plug-in y el método Cross-validation. Este último procedimiento será descrito más adelante, para el caso de regresión con Kernels. Para mayor detalle de estos procedimientos ver Hardle (1991) y Park y Turlach (1992).

2.3. Regresión con Kernels basado en el estimador de Nadaraya Watson

En la sección anterior se ha presentado la estimación de $f(x)$ basada en funciones Kernel. Para poder estimar el valor $m(x)$, es necesario determinar ahora la estimación de $f(x, y)$ (ver 4). Siguiendo el procedimiento en (9) se tiene:

$$\begin{aligned}
\int y\hat{f}_{h_1,h_2}(x,y)dy &= \frac{1}{n}\sum_{i=1}^{n} K_{h_1}(x-X_i)\int yK_{h_2}(y-Y_i)dy \\
&= \frac{1}{n}\sum_{i=1}^{n} K_{h_1}(x-X_i)\int \frac{y}{h_2}K(\frac{y-Y_i}{h_2})dy \\
&= \frac{1}{n}\sum_{i=1}^{n} K_{h_1}(x-X_i)\int (sh_2+Y_i)K(s)ds \\
&= \frac{1}{n}\sum_{i=1}^{n} K_{h_1}(x-x_i)Y_i
\end{aligned} \tag{21}$$

Reemplazando (9) y (5) en la expresión (4) y usando el mismo ancho h; se obtiene la siguiente estimación de la función $m(x)$:

$$\hat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x-X_i)Y_i}{\sum_{i=1}^{n} K_h(x-X_i)} \tag{22}$$

Es decir, a partir de (22) y (5) los pesos W_i se pueden obtener como:

$$W_{hi}(x) = \frac{K_h(x-X_i)}{n^{-1}\sum_{i=1}^{n} K_h(x-X_i)} = \frac{K_h(x-X_i)}{\hat{f}_h(x)} \tag{23}$$

El subíndice h indica la dependencia de los pesos con el ancho de banda usado en la función Kernel. Al estimador $\hat{m}_h(\bullet)$ en (5) se le conoce como estimador Nadaraya-Watson (Nadaraya, 1964; Watson, 1964). Este estimador puede ser visto como un promedio (local) ponderado de la variable respuesta Y_i. La función Kernel en (22) puede ser cualquiera de los Kernel antes mencionados. Los pesos W_{hi} verifican los siguientes puntos:

- Dependen de toda la muestra $\{X_i\}_{i=1}^{n}$ a través del denominador que es una estimación de la función de densidad.

- Cuando $h \to 0$, $W_{hi}(x) \to n$ si $x = X_i$ (y es único) entonces la estimación en X_i converge al valor Y_i, es decir, es una interpolación de los datos.
- Cuando $h \to \infty$, $W_{hi}(x) \to 1 \ \forall x$, por lo que $\hat{m}_h(x) \to \bar{y}$.
- h determina el nivel de suavidad del estimador $\hat{m}_h(x)$, de la misma forma que en la estimación de densidades.

En la figura (5(a)) se representan dos estimaciones de m. Ambas estimaciones se basan en $n = 200$ valores generados como $Y_i = \sin^3(2\pi X_i)^3 + \varepsilon_i$, estimador de Nadaraya-Watson con Kernel Gaussiano y dos anchos de banda, $h = 0,025$ y $h = 0,06$. Hay que tener en cuenta que el ajuste se ha realizado dentro de la muestra, es decir, en la estimación de $m(x)$ se ha utilizado el punto (x, y). Para poder evaluar el ajuste de una función es necesario estimar el error fuera de la muestra, es decir, estimar $m(x)$ sin utilizar en la estimación el punto (x, y). Igual que en la estimación de densidades, el parámetro h influye mucho en el ajuste de la función. La selección de h es muy importante y será explicada en otra sección de este capítulo. En la figura (5(b)) se muestra el ajuste usando dos Kernel distintos: Gaussiano y Quartic, con $h = 0,06$. En Hardle (1990) se concluye que la selección del tipo de Kernel no es crítica en el ajuste, sin embargo, la selección del parámetro h sí lo es.

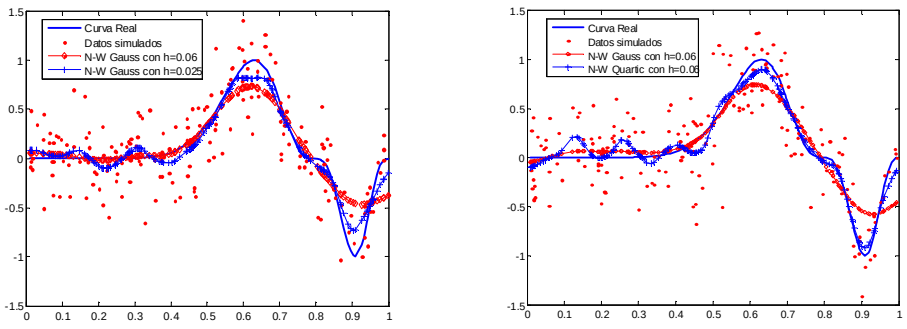


Figura 5: (a) Ajuste usando Nadaraya-Watson basado en Kernel Gaussiano y h=0,025 y h=0,06. (b) Ajuste usando Nadaraya-Watson basado en Kernels Gaussiano y Quartic, con h=0,06.

2.4. Regresión polinómica local

El estimador de Nadaraya-Watson es una forma de regresión no paramétrica basada en Kernels. Un método más general de estimación basada en Kernels, es el conocido como regresión polinómica local. Este método permite aproximar una curva $m(\bullet)$ mediante polinomios

de grado p, que son ajustados usando mínimos cuadrados ponderados (WLS). En realidad, el estimador de Nadaraya-Watson es un caso especial de regresión polinómica correspondiente a un polinomio de grado cero. Por ejemplo, para un x dado, el estimador $\hat{m}_h(x)$ en (22), con pesos W_{hi}, puede ser escrito como la solución del siguiente problema de minimización:

$$\min_{\beta_0} \sum_{i=1}^{n} W_{hi} \left(Y_i - \beta_0\right)^2 = \sum_{i=1}^{n} W_{hi} \left[Y_i - \hat{m}_h(x)\right]^2 \tag{24}$$

donde β_0 es un polinomio de grado cero. Resolviendo mediante la primera derivada con resppecto a β_0 se tiene:

$$\sum_{i=1}^{n} W_{hi} \left(Y_i - \beta_0\right) = 0$$

$$\hat{\beta}_0(x) = \hat{m}_h(x) = \frac{1}{n} \sum_{i=1}^{n} W_{hi} Y_i$$

donde se ha considerado, a partir de (23), que $\sum_{i=1}^{n} W_{hi} = n$.

La función $m(\cdot)$ puede, sin embargo, ser aproximada por un polinomio de grado p. Basados en el desarrollo de Taylor la función $m(t)$ puede ser aproximada como:

$$m(t) \approx m(x) + m'(x)(t-x) + ... + m^{(p)}(x)(t-x)^p \frac{1}{p!} \tag{25}$$

donde t, es un punto en la vecindad de x. Esta aproximación sugiere la posibilidad de una regresión polinómica local en el entorno de x. La función Kernel $K_h(x - X_i)$ puede ser incluida como peso (en lugar de los pesos W_{hi} del estimador Nadaraya-Watson), de manera que se ajuste un polinomio teniendo en cuenta la proximidad al punto x. Siguiendo el planteamiento hecho en (24), el problema queda planteado como:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} K_h(x - X_i) \left[Y_i - \beta_0 - \beta_1(X_i - x) - ... - \beta_1(X_i - x)^p\right]^2 \tag{26}$$

Aquí $\boldsymbol{\beta}$ representa el vector de coeficientes $(\beta_0, \beta_1, ..., \beta_p)'$. La solución del problema, es decir la estimación de $\boldsymbol{\beta}$, puede ser obtenida aplicando mínimos cuadrados ponderados. Denotando como:

$$\mathbb{X} = \begin{pmatrix} 1 & X_1 - x & (X_1 - x)^2 & ... & (X_1 - x)^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_n - x & (X_n - x)^2 & ... & (X_n - x)^p \end{pmatrix} \tag{27}$$

$$\mathbf{Y} = (Y_1, ...Y_n)' \tag{28}$$

$$\mathbf{W} = \begin{pmatrix} K_h(x - X_1) & ... & 0 \\ 0 & ... & 0 \\ \vdots & ... & \vdots \\ 0 & ... & K_h(x - X_n) \end{pmatrix} \tag{29}$$

el estimador $\hat{\boldsymbol{\beta}}$ es igual a:

$$\hat{\beta}(x) = (\mathbb{X}'\mathbf{W}\mathbb{X})^{-1}\mathbb{X}'\mathbf{W}\mathbf{Y} \tag{30}$$

Es necesario destacar que en contraste con el ajuste paramétrico de mínimos cuadrados, el estimador $\hat{\beta}$ varía con x. El estimador de la función $m(\cdot)$ en el punto x obtenido mediante regresión polinómica local, es expresado como:

$$\hat{m}_{p,h}(x) = \hat{\beta}_0(x) \tag{31}$$

ya que a partir de (25) y (26) se tiene que $m(x) \approx \beta_0(x)$. La estimación de toda la curva $\hat{m}_{p,h}(\cdot)$ se obtiene mediante regresión polinómica local para distintos valores x. El estimador $\hat{m}_{p,h}$ depende del grado del polinomio p y, del parámetro h a través de la función Kernel. Este parámetro h es un parámetro de suavizado de la función, de la misma forma que en el estimador de Nadaraya Watson. En la figura (6) se representa el ajuste de los datos usados en la figura (5). Las tres curvas representadas son la curva real, ajuste con Nadaraya Watson y ajuste con regresión polinómica de grado $p = 3$. El Kernel usado en ambas estimaciones es el Gaussiano y el ancho de banda es $h = 0,06$. En este caso, se obtiene un mejor ajuste mediante regresión polinómica local.

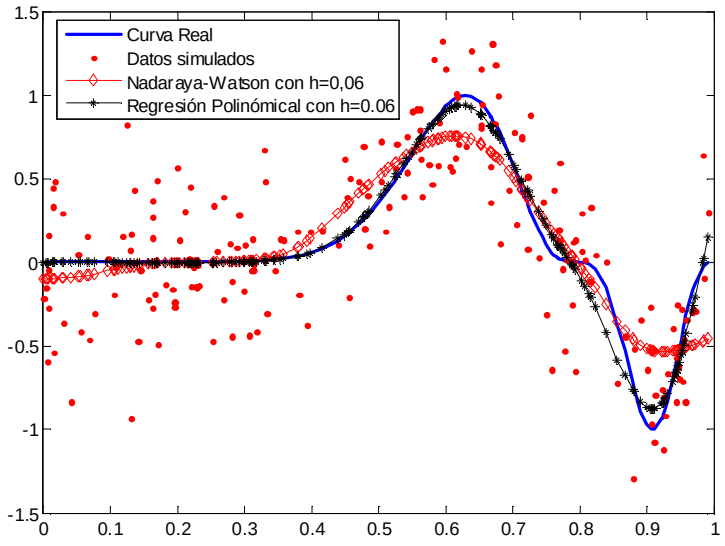


Figura 6: Ajuste usando Nadaraya-Watson y Regresión Polinómica, ambos basado en Kernel Gaussiano y h=0,06

2.4.1. Selección del parámetro de suavizado h

Igual que para la estimación de funciones de densidad, en la regresión basada en Kernels es importante la selección del parámetro de suavizado h. También, al igual que en la estimación de densidades, el parámetro h puede ser seleccionado minimizando ciertos parámetros que miden el grado de ajuste de la curva, por ejemplo el MSE, AMISE, etc. A continuación se describen brevemente algunos de estos parámetros, y los dos procedimiento más habituales para la selección del parámetro h: Cross-validation y Funciones de penalización.

1. *Error cuádratico medio (MSE)*

$$\text{MSE}\left\{\hat{m}_h(x)\right\} = E\left[\left\{\hat{m}_h(x) - m(x)\right\}^2\right] \tag{32}$$

Aquí $m(x)$ es una constante pero $\hat{m}_h(x)$ es una variable aleatoria, por tanto, el valor esperado de MSE depende de la distribución de ésta variable. El estimador $\hat{m}_h(x)$ depende de los datos $\{X_i, Y_i\}_{i=1}^n$, luego MSE

$\{\hat{m}_h(x)\}$=$\int ... \int \{\hat{m}_h(x) - m(x)\} f(x_1, ..., x_n, y1, ..., y_n) dx_1 ... dx_n dy_1 ... dy_n$.

Una desventaja importante del MSE es que se trata de una medida del error de la estimación de m en un punto x. Es necesario, por tanto, definir una medida global del error de la estimación.

2. *Error cuadrático integrado (ISE-integrated squared error)*

El ISE es una medida global del error de estimación y se define como:

$$\text{ISE}(h) = \text{ISE}\left\{\hat{m}_h\right\} = \int_{-\infty}^{\infty} \left\{\hat{m}_h(x) - m(x)\right\}^2 w(x) f_X(x) dx \tag{33}$$

El estimador $\hat{m}_h(x)$ depende de los datos utilizados en la estimación, por tanto, el ISE es también una variable aleatoria. La función de pesos $w(\cdot)$ en (33) puede ser utilizada para asignar menos peso a las observaciones en regiones que existe mucha dispersión de datos o en los tallos de la distribución.

3. *Error cuadrático medio integrado (MISE-mean integrated squared error)*

$$\begin{aligned} \text{MISE}(h) &= \text{MISE}\left\{\hat{m}_h\right\} = E\left\{\text{ISE}(h)\right\} \\ &= \int ... \int \left[\int_{-\infty}^{\infty} \left\{\hat{m}_h(x) - m(x)\right\}^2 w(x) f_X(x) dx\right] \cdot \\ &\quad \cdot f(x_1, ..., x_n, y_1, ..., y_n) dx_1 ... dx_n dy_1 ... dy_n \end{aligned} \tag{34}$$

En este caso, no se trata de una variable aleatoria. Se trata del valor esperado del valor de la variable ISE respecto a todos lo posibles valores de X e Y.

4. *Error cuadrático promedio (ASE-averaged squared error)*

$$\text{ASE}(h) = \text{ASE}\left\{\hat{m}_h\right\} = \frac{1}{n}\sum_{j=1}^{n}\left\{\hat{m}_h(X_j) - m(X_j)\right\}^2 w(X_j) \tag{35}$$

Se trata de una aproximación discreta del ISE. Al igual que éste el ASE es una variable aleatoria de la medida de discrepancia.

5. *Error medio cuadrático promedio (MASE- mean averaged squared error)*

$$\text{MASE}(h) = \text{MASE}\left\{\hat{m}_h\right\} = E\left\{\text{ASE}(h)|X_1 = x_1, ..., X_n = x_n\right\} \tag{36}$$

Es la esperanza condicionada del ASE. Usa la función de densidad conjunta de $Y_1, Y_2, ..., Y_n$. Si $X_1, ..., X_n$ es una variable aleatoria, MASE también lo es.

¿Qué medida de discrepancia se debería utilizar para la elección de un ancho de banda h adecuado?

Basados en la estimación de densidades mediante Kernels, la elección natural es utilizar el MISE o su versión asintótica AMISE. El AMISE, sin embargo, tiene más parámetros desconocidos en el caso de regresión que en el caso de densidades.

Con respecto al procedimiento para evaluar estos parámetros, las técnicas más utilizadas son: Cross-Validation y Funciones de Penalización. A continuación se presenta un ejemplo para ilustrar estos procedimientos en el caso del estimador de Nadaraya-Watson. En el caso de este estimador, se ha demostrado en Marron & Härdle (1986) que el ASE, ISE y MISE producen de manera asintótica el mismo nivel de suavizado. Por ello, se usará el ASE por ser éste el criterio más fácil de utilizar.

Cross-validation y Funciones de penalización Se trata de encontrar el valor de h que minimice el ASE(h) :

$$\text{ASE}(h) = \frac{1}{n}\sum_{i=1}^{n} m^2(X_i)w(X_i) + \frac{1}{n}\sum_{i=1}^{n}\hat{m}_h^2(X_i)w(X_i) - 2\frac{1}{n}\sum_{i=1}^{n} m(X_i)\hat{m}_h(X_i)w(X_i)$$

Dado que el ASE es una variable aleatoria, tomamos su valor esperado condicionado a la muestra, MASE(h) :

$$\begin{aligned}
\text{MASE}(h) &= E\left\{\text{ASE}(h)|X_1 = x_1, ..., X_n = x_n\right\} \qquad (37)\\
&= \frac{1}{n}\sum_{i=1}^{n} E\left[\left\{\hat{m}_h(X_i) - m(X_i)\right\}^2 |X_1 = x_1, ..., X_n = x_n\right] w(X_i)\\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\begin{array}{l}\text{Var}\left\{\hat{m}_h(X_i)|X_1 = x_1, ..., X_n = x_n\right\} +\\ \text{Sesgo}\left\{\hat{m}_h(X_i)|X_1 = x_1, ..., X_n = x_n\right\}\end{array}\right] w(X_i)\\
&= \frac{1}{n}\sum_{i=1}^{n}\left[v(h) + b^2(h)\right] w(X_i)
\end{aligned}$$

donde,

$$b^2(h) = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{1}{n}\sum_{j=1}^{n}\frac{K_h\left(X_i - X_j\right)}{\hat{f}_h(X_i)}m(X_j)\right\}^2 w(X_i)$$

y,

$$v(h) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{n^2}\sum_{j=1}^{n}\left\{\frac{K_h(X_i - X_j)}{\hat{f}_h(X_i)}\right\}^2 \sigma(X_j)^2\right] w(X_i)$$

A partir de la expresión (37) se puede concluir que el valor óptimo de h está relacionado con la varianza y el cuadrado del sesgo. Además, se observa que el valor de MASE en (37) depende de la función $m(\bullet)$, que es desconocida. Este valor de $m(\bullet)$ puede ser reemplazado por los valores Y_i observados (valores en la muestra utilizada en la estimación). Es decir, utilizar en (37) la siguiente expresión:

$$p(h) = \frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \hat{m}_h(X_i)\right\}^2 w(X_i) \tag{38}$$

que es una suma ponderada del cuadrado de los residuos. Un problema de utilizar Y_i es que este valor es también usado en la estimación de $\hat{m}_h(X_i)$. Como consecuencia, el valor $p(h)$ puede ser arbitrariamente más pequeño haciendo $h \to 0$ (en cuyo caso, el valor estimado $\hat{m}(\bullet)$ es una interpolación de los Y_i). Con el fin de analizar mejor el valor de $p(h)$, podemos sumar y restar a (38) el valor $m(X_i)$:

$$\begin{aligned} p(h) &= \frac{1}{n}\sum_{i=1}^{n}\left[\left\{Y_i - m(X_i)\right\} + \left\{m(X_i) - \hat{m}_h(X_i)\right\}\right]^2 w(X_i) && (39)\\ &= \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2 w(X_i) + ASE(h) - \frac{2}{n}\sum_{i=1}^{n}\varepsilon_i\left\{\hat{m}_h(X_i) - m(X_i)\right\} w(X_i) && (40)\end{aligned}$$

donde $\varepsilon_i = Y_i - m(X_i)$. Notar que el primer término $\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2 w(X_i)$ no depende de h y el segundo término es el ASE(h). Notar también que el tercer término tiende a cero a medida que la varianza tiende a cero y tiene signo negativo. Luego, el valor $p(h)$ es un estimador sesgado a la baja del ASE(h), y por consiguiente, la selección del parámetro h usando $p(h)$ es una estimación sesgada negativamente del valor h que minimiza el ASE(h).

En la práctica se utilizan dos formas para tratar este problema: Funciones de Penalización y Cross-validation.

1. *Funciones de Penalización*

   Mediante esta técnica se multiplica el valor $p(h)$ por una función de penalización que corrige el sesgo negativo. Algunas de las funciones de penalización utilizadas son las siguientes: Shibata's model selector (Shibata, 1981), Generalized cross-validation (Craven and Wahba, 1979; Li, 1985), Akaike's Information Criterion (Akaike, 1970), Finite Prediction Error (Akaike, 1974) y Rice's (Rice, 1984).

2. *Cross-Validation*

Mediante esta técnica se utiliza en (38) el estimador $\hat{m}_{h,-1}(X_i)$, denominado "leave-one-out", en lugar de $\hat{m}_h(X_i)$. En el caso del estimador Nadraya-Watson, se tiene entonces:

$$\hat{m}_{h,-i}(X_i) = \frac{\sum_{j\neq i} K_h(X_i - X_j)Y_j}{\sum_{j\neq i} K_h(X_i - X_j)} \tag{41}$$

En este caso, al estimar $\hat{m}_h(\cdot)$ en X_i no se utiliza la observación i-ésima (como indica el subíndice $-i$). Luego, la función Cross-Validation es:

$$CV(h) = \sum_{i=1}^{n} \{Y_i - \hat{m}_{h,-i}(X_i)\}^2 w(X_i) \tag{42}$$

Al usar $\hat{m}_{h,-i}(X_i)$, el tercer término de la expresión (39) es igual a cero:

$$E\left[-\frac{2}{n}\sum_{i=1}^{n} \varepsilon_i \{\hat{m}_{h,-i}(X_i) - m(X_i)\} w(X_i) | X_1 = x_1, ..., X_n = x_n\right] = 0 \tag{43}$$

Además, el primer término no depende de h, luego, minimizar (42) es, en media, igual que minimizar el valor ASE(h).Luego, se puede concluir que seleccionar el valor de h que minimice $CV(h)$ es una regla teóricamente adecuada y aplicable en la práctica.

Las técnicas de estimación descritas hasta ahora se han centrado en el caso univariante. En la siguiente sección se extiende el uso de funciones Kernel al caso multivariante.

2.5. Regresión polinómica local multivariante

2.5.1. Estimación de densidades

En el caso multivariante se tiene un conjunto de d variables $\mathbf{X} = (X_1, X_2, ..., X_d)'$. Siguiendo el caso univariante, la estimación de la función de densidad basada en Kernel es ente caso:

$$\hat{f}_{\mathbf{h}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_1...h_d} \mathcal{K}\left(\frac{x_1 - X_{i1}}{h_1}, ..., \frac{x_d - X_{id}}{h_d}\right) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{K}_{\mathbf{h}}(\mathbf{x} - \mathbf{X}_i) \tag{44}$$

Aquí, $\mathbf{x} = (x_1, ..., x_d)'$, $\mathbf{X}_i = (X_{1i}, ..., X_{di})'$, $\mathbf{h} = (h_1, .., h_d)'$ y $\mathcal{K}(\bullet)$ representa la función Kernel multivariante. La forma más sencilla de $\mathcal{K}(\cdot)$ es considerar un Kernel multiplicativo, es decir:

$$\mathcal{K}(u) = K(u_1) \cdot ... \cdot K(u_d)$$

donde $K(\cdot)$ representa una función Kernel univariante, como las descritas en las secciones anteriores. En este caso, la expresión (44) es igual a:

$$\hat{f}_{\mathbf{h}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \left\{\prod_{j=1}^{d} h_j^{-1} K\left(\frac{x_j - X_{ij}}{h_j}\right)\right\} = \frac{1}{n}\sum_{i=1}^{n} \left\{\prod_{j=1}^{d} K_{h_j}(x_j - X_{ij})\right\} \tag{45}$$

Otra alternativa para $\mathcal{K}(\cdot)$ es utilizar una función mutivariante, por ejemplo, el Kernel multivariante de Epanechnikov:

$$\mathcal{K}(\mathbf{u}) \infty (1 - \mathbf{u}'\mathbf{u}) I(\mathbf{u}'\mathbf{u} \leq 1)$$

donde ∞ significa proporcional.

En el caso del vector de anchos de banda $\mathbf{h}$, es posible simplificarlo, utilizando el mismo ancho de banda h para cada variable X_i. En ese caso el estimador $\hat{f}_{\mathbf{h}}(\mathbf{x})$ en (45) sería igual a:

$$\hat{f}_{\mathbf{h}}(\mathbf{x}) = \frac{1}{nh^j} \sum_{i=1}^{n} \left\{ \prod_{j=1}^{d} K\left(\frac{x_j - X_{ij}}{h} \right) \right\} = \frac{1}{n} \sum_{i=1}^{n} \left\{ \prod_{j=1}^{d} K_h \left(x_j - X_{ij} \right) \right\}$$

Por el contrario, una forma mucho más general de definir el ancho de banda, es usar una matriz $\mathbf{H}$ (en lugar de un vector). Esta matriz $\mathbf{H}$ es una matriz no singular. En este caso, el estimador de la función de densidad en (44)sería igual a:

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\det(\mathbf{H})} \mathcal{K}\left\{ \mathbf{H}^{-1}(\mathbf{x} - \mathbf{X}_i) \right\} = \frac{1}{n} \sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i) \tag{46}$$

El uso de $\mathbf{H} = \widehat{\mathbf{\Sigma}}^{-1/2}$, donde $\widehat{\mathbf{\Sigma}}$ es la matriz de covarianza de los datos, es una buena regla para optimizar el ancho de banda del modelo. La expresión (46) engloba, en realidad los tres casos. El caso de $\mathbf{h} = (h_1, .., h_d)'$ es igual a usar $\mathbf{H}$ =diag$(h_1, .., h_d)$. El caso de iguales anchos de banda h, equivale a usar $\mathbf{H}$ =$h\mathbf{I}_{d.}$, donde $\mathbf{I}$ es la matriz identidad $d \times d$. matriz

2.5.2. Estimador Multivariante de Nadaraya-Watson

En el caso multivariante es estimador de $f(y, \mathbf{X})$ será igual a:

$$\hat{f}_{h,\mathbf{H}}(y, \mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_h(Y_i - y) \mathcal{K}_{\mathbf{H}}(\mathbf{X}_i - \mathbf{x}) \tag{47}$$

donde el subíndice $\mathbf{H}$ hace referencia a la matriz de ancho de banda definido para las variables $X_1, ..., X_d$, y el subíndice h es el ancho de banda para la variable Y. El estimador de Nadaraya-Watson $\hat{m}_{\mathbf{H}}(\mathbf{x})$multivariante es entonces igual a:

$$\hat{m}_{\mathbf{H}}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(\mathbf{X} - \mathbf{x}) Y_i}{\sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(\mathbf{X} - \mathbf{x})} \tag{48}$$

2.5.3. Regresión polinómica local multivariante

El modelo de regresión polinómica local en el caso multivariante es una generalización del caso univariante. El problema de minimización, por ejemplo en el caso de polinomios de grado 1 (lineales) es igual a:

$$\min_{\beta_0, \boldsymbol{\beta}_1} \sum_{i=1}^{n} \mathcal{K}_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i) \left[Y_i - \beta_0 - \boldsymbol{\beta}_1'(\mathbf{X}_i - \mathbf{x}) \right]^2$$

La solución de este problema es:

$$\hat{\boldsymbol{\beta}} = (\beta_0, \boldsymbol{\beta}_1')' = (\mathbb{X}^{'}\mathbf{W}\mathbb{X})^{-1}\mathbb{X}'\mathbf{W}\mathbf{Y} \tag{49}$$

donde $\hat{\boldsymbol{\beta}} = (\beta_0,\beta_{11},...,\beta_{1d})'$, $\mathbf{Y} = (Y_1, ...Y_n)'$,

$$\mathbb{X} = \begin{pmatrix} 1 & (\mathbf{X}_1 - \mathbf{x})' \\ \vdots & \vdots \\ 1 & (\mathbf{X}_n - \mathbf{x})' \end{pmatrix} = \begin{pmatrix} 1 & X_{11} - x_1 & ... & (X_{1d} - x_d) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} - x_1 & ... & (X_{nd} - x_d) \end{pmatrix}$$

siendo $\mathbb{X}$ de dimensión $n \times (d + 1)$ y:

$$\mathbf{W} = \begin{pmatrix} \mathcal{K}_{\mathbf{H}}(\mathbf{X}_1 - \mathbf{x}) & ... & 0 \\ \vdots & ... & \vdots \\ 0 & ... & \mathcal{K}_{\mathbf{H}}(\mathbf{X}_n - \mathbf{x}) \end{pmatrix}$$

El estimador de la función $m(\cdot)$ puede ser obtenido, siguiendo el caso univariante, como:

$$\hat{m}_{1,\mathbf{H}}(\mathbf{x}) = \hat{\beta}_0(\mathbf{x})$$

El problema se puede generalizar fácilmente a regresión polinómica de grado p. En este caso, el vector $\mathbf{Y}$ y la matriz $\mathbf{W}$ son iguales. El vector de coeficientes beta será igual a $\hat{\boldsymbol{\beta}} = (\beta_0,\beta_{11},...,\beta_{1d},...,\beta_{p1},...,\beta_{pd})'$, y la matriz $\mathbb{X}$, que tendrá dimensión $n \times [(d \times p) + 1]$, será igual a:

$$\mathbb{X} = \begin{pmatrix} 1 & (\mathbf{X}_1 - \mathbf{x})' & ... & [(\mathbf{X}_1 - \mathbf{x})^p]' \\ \vdots & \vdots & \vdots & \\ 1 & (\mathbf{X}_n - \mathbf{x})' & ... & [(\mathbf{X}_1 - \mathbf{x})^p]' \end{pmatrix}$$

En la figura (7) se representa con puntos "•ün conjunto de $n = 200$ datos obtenidos según la relación $Y = m(\mathbf{X})$ =sen$(2\pi X_1)+X_2$, donde X_i, $i = 1, 2$ está uniformemente distribuida en $[0, 1]$. Se ha estimado la función $\hat{m}_{1,\mathbf{H}}(\bullet)$ utilizando regresión polinómica local de grado $p = 3$, Kernel Gaussiano y $h_1 = h_2 = h = 0,08$. En la estimación se han utilizado $n = 200$ datos Y_1, generados como Y_1 =sen$(2\pi X_1) + X_2 + \varepsilon_i$ donde $\varepsilon_i \sim N(0, 0{,}05)$. Los valores estimados mediante regresión polinómica local se representan con "*". Observando la figura (7) se puede concluir que el modelo permite un buen ajuste de la función $m(\bullet)$. Hay que tener en cuenta, sin embargo que se ha realizado una estimación dentro de la muestra y sería necesario, evaluar el modelo fuera de la muestra.

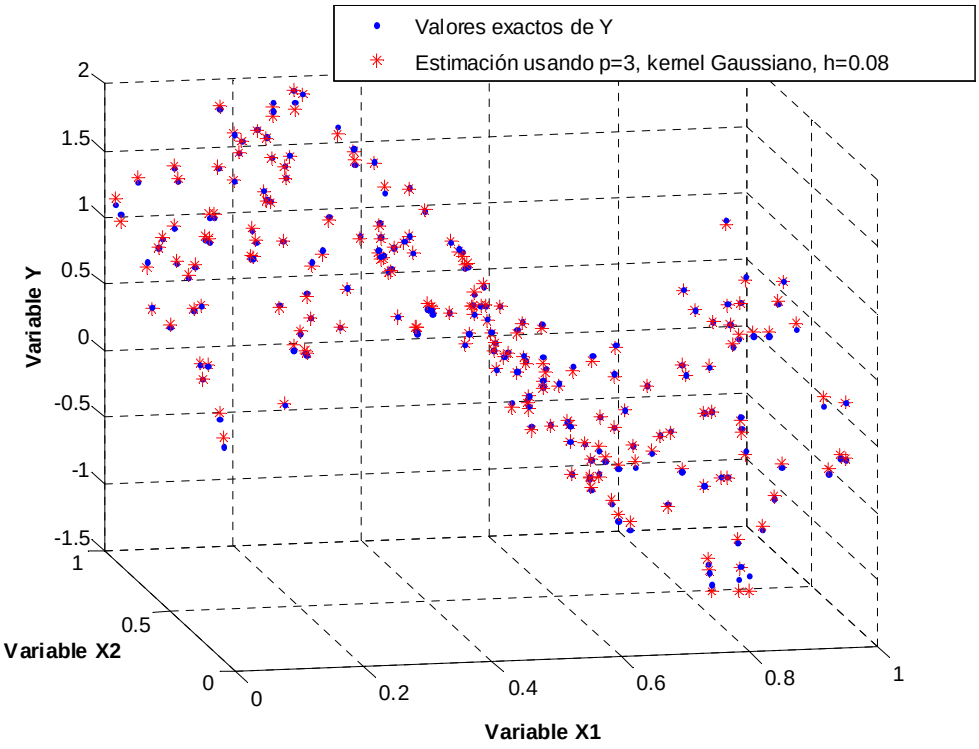


Figura 7: Ajuste mediante regresión polinómica multivariante usando p=3, Kernel Gaussiano y h=0,08

3. GUI de Matlab usada para desarrollar la Interfaz

3.1. Acerca de Matlab

MATLAB, nombre abreviado de "MATrix LABoratory", es un programa muy potente para realizar cálculos numéricos con vectores y matrices. Como caso particular puede trabajar también con números escalares, tanto reales como complejos. Una de las capacidades más atractivas que ofrece es la de realizar una amplia variedad de gráficos en dos y tres dimensiones (aunque este último no se va a explotar para la realización de este proyecto).

MATLAB se utiliza ampliamente en:

- Cálculos numéricos
- Desarrollo de algoritmos
- Modelado, simulación y prueba de prototipos
- Análisis de datos, exploración y visualización
- Graficación de datos con fines científicos o de ingeniería
- Desarrollo de aplicaciones que requieran de una interfaz gráfica de usuario (GUI, Graphical User Interface).

El entorno de trabajo de Matlab Guide (GUI) es muy gráfico e intuitivo, como muestra la figura (8) Algunos de los componentes más importantes del entorno de trabajo de Matlab son el editor de caminos de búsqueda (Path Browser), la ventana de comandos (Command window) y el editor y depurador de errores (Editor & Debugger).

En la figura (8) aparece señalada en la zona izquierda la herramienta GUIDE (GUI Builder) dentro de las funciones de Matlab. GUI son las siglas anglosajonas cuya traducción es la de Interfaz Gráfica para el Usuario, y va a ser la herramienta utilizada para la construcción de este proyecto. Más adelante se comentará más detenidamente sus funciones así como las aplicaciones que tiene.

A continuación, se describen algunas de las herramientas propias de Matlab que en numerosas ocasiones han sido utilizadas:

3.1.1. Editor de caminos de búsqueda

Matlab puede llamar a una gran variedad de funciones, tanto propias como programadas por los usuarios. Por tanto, conviene saber donde se quiere ejecutar la función o fichero *.m. Con el editor de caminos se puede buscar la carpeta que se necesita, si es que se desconoce su ubicación (search path), o bien se puede elegir la carpeta, donde se encuentra el fichero, de una lista que contiene todos los directorios (path browser).

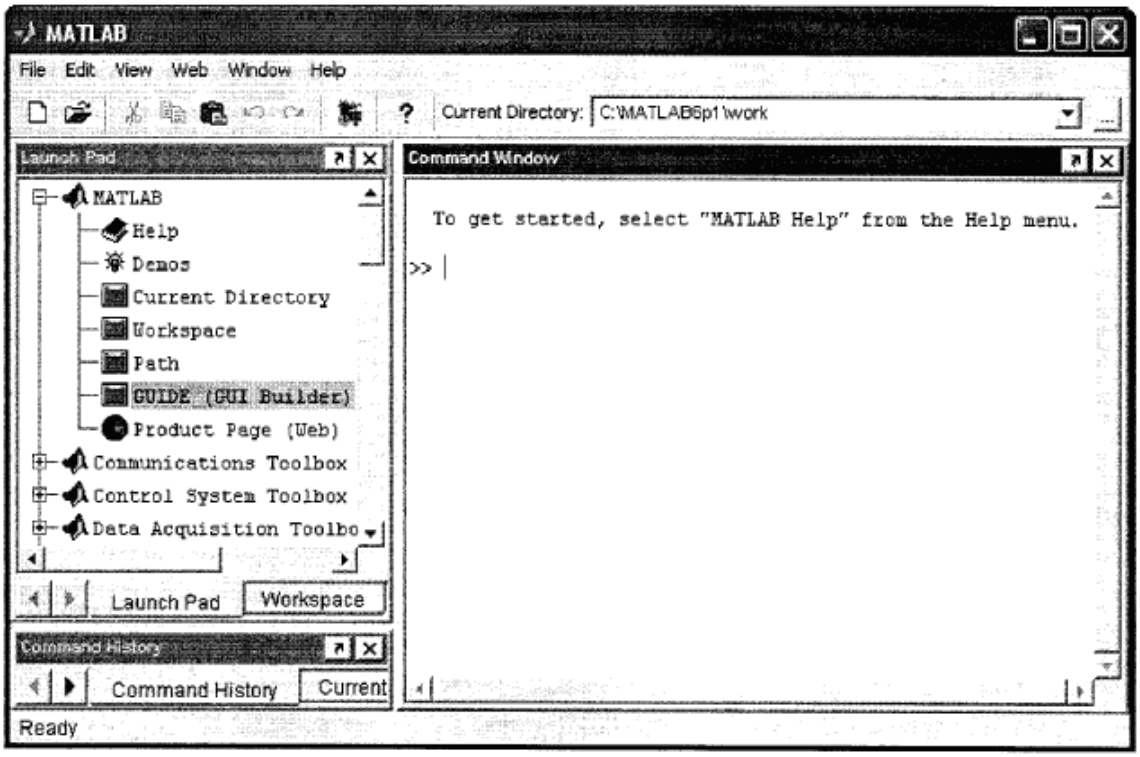


Figura 8: Pantalla principal de MatLab

3.1.2. Ventana de comandos

Es la ventana principal de trabajo de Matlab, donde se puede llamar a funciones o ficheros, se pueden escribir todos los comandos posibles que permite Matlab, se puede acceder a la librería de ayuda de Matlab, se muestran las soluciones de cálculo y, entre otras muchas funciones, se indican los errores producidos al ejecutar una función.

3.1.3. Depurador de errores.

En Matlab tiene particular importancia los ficheros-M . La importancia de estos ficheros-M es que al teclear su nombre en la línea de comandos y pulsar intro se ejecutan uno tras otro todos los comandos contenidos en dicho fichero. Además de esto, MatLab permite depurar los errores que pueda contener el archivo con extensión *.m mediante la opción de ejecutar por fragmentos.

3.2. Interfaz gráfica. GUI

Matlab permite desarrollar fácilmente un conjunto de pantallas (paneles) con botones, menús, ventanas, etc., que permiten utilizar de manera muy simple programas realizados

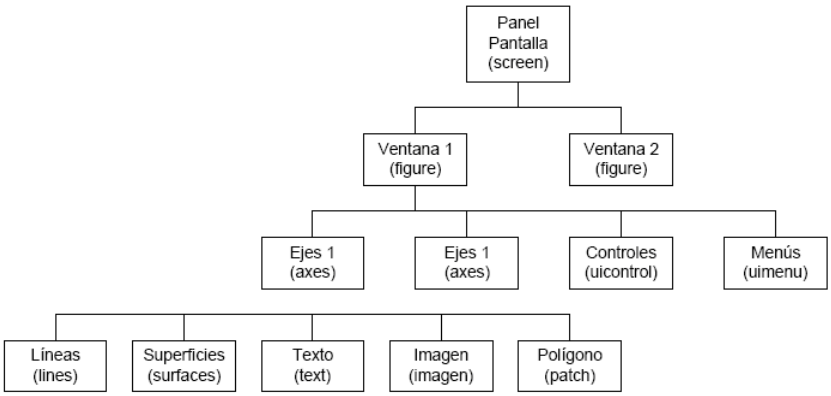


Figura 9: Jerarquía gráfica en MatLab

dentro de este entorno. Este conjunto de herramientas se denomina **interfaz gráfica de usuario (GUI)**. Las posibilidades que ofrece MATLAB no son muy amplias, en comparación a otras aplicaciones de Windows como Visual Basic, Visual C. La elaboración de GUIs puede llevarse a cabo de dos formas, la primera de ellas consiste en escribir un programa que genere la GUI (script), la segunda opción consiste en utilizar la herramienta de diseño de GUIs, incluida en el Matlab, llamada **GUIDE**. En esta sección se abordarán ambas formas de crear GUIs. Para poder hacer programas que utilicen las capacidades gráficas avanzadas de MATLAB hay que conocer algunos conceptos que se explican en los siguientes apartados

El panel GUI se crea en una ventana, identificada como figura y está formada por los siguientes elementos:

- Menú de interfaz con el usuario
- Dispositivos de control de la interfaz con el usuario
- Ejes para desplegar las gráficas o imágenes

3.2.1. Estructura de gráficos en Matlab

Los gráficos de MATLAB tienen una estructura jerárquica formada por objetos de distintos tipos. Esta jerarquía tiene forma de árbol, con el aspecto mostrado en la figura (9).

3.2.2. Objetos gráficos en Matlab

Según se muestra en la figura (9), el objeto más general es la pantalla o panel (*screen*). Dicho objeto es la raíz de todos los demás y sólo puede haber un objeto pantalla. Una pantalla

puede contener una o más ventanas (*figures*). A su vez cada una de las ventanas puede tener uno o más ejes de coordenadas (*axes*) en los que se puede representar otros objetos de más bajo nivel. Una ventana puede tener también controles (*uicontrols*) tales como botones, barras de desplazamiento, botones de selección o de opción, etc.) y menús (*uimenus*). Finalmente, los ejes pueden contener los cinco tipos de elementos gráficos que permite MATLAB: líneas (*line*), polígonos (*patch*), superficies (*surfacs*), imágenes (image) y texto (*text*). La jerarquía de objetos mostrada en la figura (9) indica que en MATLAB hay "objetos padres e hijos". Por ejemplo, todos los objetos ventana son hijos de pantalla, y cada ventana es padre de los objetos ejes, controles o menús que están por debajo. A su vez los elementos gráficos (líneas, polígonos, etc.) son hijos de un objeto ejes, y no tienen otros objetos que sean sus hijos.

Cuando se borra un objeto de MATLAB automáticamente se borran todos los objetos que son sus descendientes.

Por ejemplo, al borrar unos ejes, se borran todas las líneas y polígonos que son hijos suyos.

3.2.3. Propiedades de los objetos

Todos los objetos de MATLAB tienen distintas propiedades. Algunas de éstas son el tipo, el estilo, el padre, los hijos, si es visible o no, y otras propiedades particulares del objeto concreto de que se trate. Las propiedades comunes a todos los objetos son: children, clipping, parent, type, UserData, Visible. Otras propiedades son propias de un tipo determinado de objeto.

Las propiedades tienen valores por omisión, que se utilizan siempre que el usuario no indique otra cosa. Es posible cambiar las propiedades por omisión, y también devolverles su valor original (llamado *factory*, por ser el valor por defecto con que salen de fábrica). El usuario puede consultar (*query*) los valores de las propiedades de cualquier objeto. Algunas propiedades pueden ser modificadas y otras no (*read only*). Hay propiedades que pueden tener cualquier valor y otras que sólo pueden tener un conjunto limitado de valores(por ejemplo, *on* y *off* en el caso de los comandos *zoom* y *grid*).

Funciones get y set MATLAB dispone de las funciones set y get para consultar y cambiar el valor de las propiedades de un objeto. Las funciones *set* (identidad) lista en pantalla todas las propiedades del objeto al que corresponde el *handle* (sólo los nombres, sin los valores de las propiedades).La función *get* (identidad) produce un listado de las propiedades y de sus valores.

3.2.4. Creación de objetos Gráficos

MATLAB permite desarrollar programas con el aspecto típico de las aplicaciones de Windows. Para todos los controles, se utilizará la función uicontrol, que permite desarrollar dichos

controles. El formato general del comando uicontrol es la mostrada en la figura (10).

```
id_control = uicontrol( id_parent,...
                'Propiedad1',valor1,...
                'Propiedad2',valor2,...
                otras propiedades
                'callback','sentencias')
```

Figura 10: Formato del comando üicontrol"

Las propiedades son las opciones del comando, que se explican en el apartado siguiente. Éstas tendrán comillas sencillas (') a su izquierda y derecha, e irán seguidas de los parámetros necesarios. En caso de que el conjunto de propiedades de un control exceda una línea de código, es posible continuar en la línea siguiente, poniendo tres puntos seguidos (. . .).

3.2.5. Controles de la interfaz gráfica de usuario

Los controles de la interfaz con el usuario en MatLab se especifican con el comando *uicontrol*. Estos controles son similares a los menús de la interfaz con el usuario, aunque los controles tienen más opciones. La sintaxis de uicontrol es la mostrada en la figura (11).

```
k=uicontrol('Style', 'especificación del estilo',...
'String', 'texto a desplegar',...
'Value', [valor],...
'BackgroundColor', [r,g,b],...
'Max', [valor],...
'Minx', [valor],...
'Position',[posición x, posición y, ancho, alto],...
'Callback', 'cadena de invocación')
```

Figura 11: Sintaxis del comando üicontrol"

donde '*especificación de estilo*' puede ser una de las siguientes cadenas (opciones): *popup, push, radio, checkbox, slider, edit* (texto editable), *text* (texto estático) o *frame*.

Texto estático (text) El texto estático puede exhibir símbolos, mensajes o incluso valores numéricos en una GUI y puede colocarse en el lugar apropiado. Este control no tiene cadena de invocación.

Menú desplegable (popup) Estos menús desplegables difieren de los menús de interfaz con el usuario en que pueden aparecer en cualquier punto del panel, mientras que los menús de interfaz con el usuario solo se encuentran en la parte superior de la ventana.

Botón (push) Los botones son pequeños objetos de la pantalla generalmente acompañados con texto. Al presionar el botón con el ratón, se producirá una acción que será ejecutada por Matlab.

Casilla de verificación (checkbox)

Las casillas están diseñadas para realizar operaciones de encendido/apagado.

Botón de radio (radio) Cuando se usa un solo botón de radio, es igual que la casilla ve verificación. Sin embargo, cuando se usan en grupo, estos son mutuamente exclusivos, es decir, si un botón de radio está encendido, los demás estarán apagados, mientras que las casillas de verificación son independientes entre sí.

Control deslizante (slider) Es un dispositivo que permite modificar un parámetro de forma continua.

Texto editable (edit) El dispositivo de texto editable le permite al usuario introducir una cadena. Puede aceptar valores numéricos en forma de vector o matriz como una cadena mediante el mismo dispositivo. La cadena de entrada puede convertirse a valores numéricos mediante la instrucción *str2num*.

Uso de varios ejes para graficación Generalmente es necesario desplegar varias gráficas dentro de una interfaz. El comando axes abre un eje (gráfica) en un punto específico dentro de un panel.

3.3. Elaboración de la Interfaz gráfica

GUIDE (Graphical User Interfase Development Environment) es un juego de herramientas que se extiende por completo el soporte de MATLAB, diseñadas para crear GUIs (Graphical User Interfaces) fácil y rápidamente dando auxilio en el diseño y presentación de los controles de la interfaz, reduciendo la labor al grado de seleccionar, tirar, arrastrar y personalizar propiedades.

Una vez que los controles están en posición se editan las funciones de llamada (Callback) de cada uno de ellos, escribiendo el código de MATLAB que se ejecutará cuando el control sea utilizado. Siempre será difícil diseñar GUIs, pero no debería ser difícil implementarlas. GUIDE está diseñado para ser menos tedioso el proceso de aplicación de la interfaz grafica y

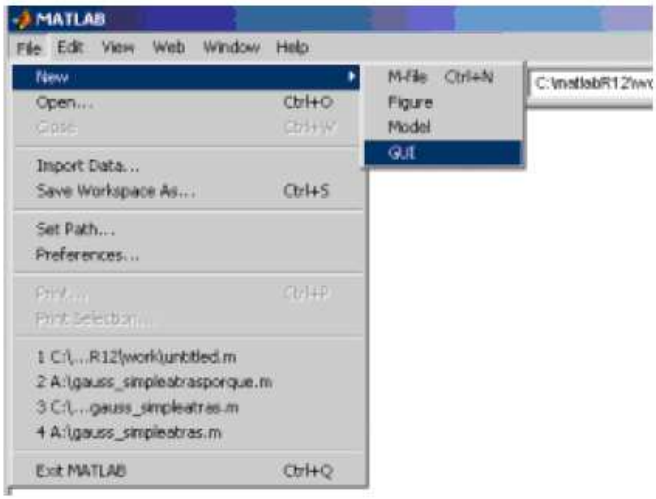


Figura 12: Iniciación de MatLab Guide

obviamente para trabajar como herramienta de trazado de GUIs, entre sus poderosos componentes esta el editor de propiedades (**property editor**), este se encuentra disponible cualquier momento que se esté lidiando con los controles de MATLAB, el editor de propiedades por separado se puede concebir como una herramienta de trazado, y asistente de codificación revisión de nombres y valores de propiedades). Cuando se fusiona con el panel de control, el editor de menú, y herramienta de alineación, resulta una combinación que brinda inigualable control de los gráficos en MATLAB.

3.3.1. Iniciación de Guide

Para ejecutar Matlab Guide, se seguirá como indica la figura (12)

3.3.2. Ventana principal

La ventana principal de Guide se compone de los elementos que se indican en la figura (13). A partir de todos ellos, el usuario lo único que debe hacer es arrastrar los elementos hasta la ventana y establecer la relación entre los mismos.

3.3.3. Flujo de operación con GUI

Con una GUI, el flujo de computo esta controlado por las acciones en la interfaz. Los comandos para crear una interfaz con el usuario se escribe en un guión, la interfaz invoca el guión que se ejecute, mientras la interfaz del usuario permanece en la pantalla aunque no se haya completado la ejecución del guión. En la figura (14) se muestra el concepto básico de

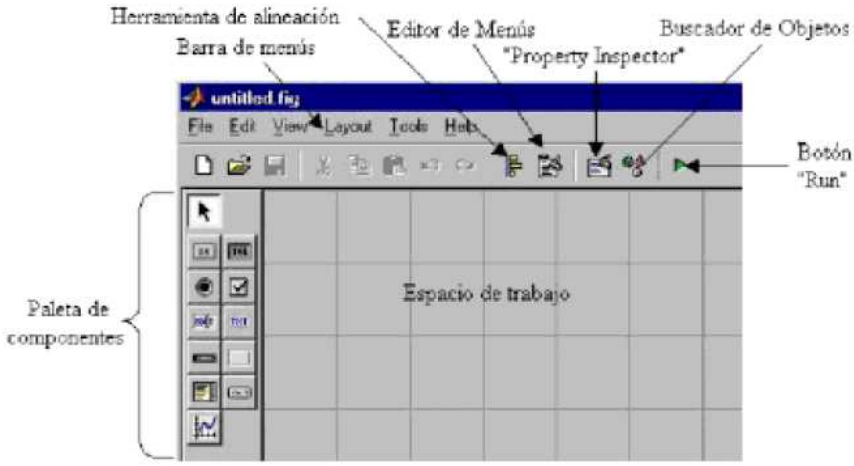


Figura 13: Principales elementos de MatLab Guide

la operación del software con una GUI. Cuando se interactúa con un control, el programa registra el valor de esa opción y ejecuta los comandos prescritas en la cadena de invocación. Los menús de interfaz con el usuario, los botones, los menús desplegables, los controladores deslizantes y el texto editable son dispositivos que controlan las operaciones del software. Al completarse la ejecución de las instrucciones de la cadena de invocación, el control vuelve a la interfaz para que puedan elegirse otra opción del menú. Este ciclo se repite hasta que se cierra la GUI.

El control guarda un string que describe la acción a realizar cuando se invoca puede consistir en un solo comando de MATLAB o una secuencia de comandos, o en una llamada a una función. Es recomendable utilizar llamadas a funciones, sobre todo cuando se requieren de más de unos cuantos comandos en la invocación.

Básicamente solo se necesita entender cinco comandos para poder describir una GUI: *uimenu, uicontrol, get, set y axes.* No obstante, lo que hace relativamente complicadas a estos comandos es el gran número de formas de uso que tiente.

3.3.4. Property Inspector

El Property Inspector (Inspector de propiedades) es una herramienta muy útil que servirá para analizar y cambiar las propiedades de cualquier elemento que componga la GUI. Es uno de los elementos más importantes para montar la interfaz, es decir, la mayor parte de operaciones se realizarán con este elemento. La función del mismo es controlar las caracteríticas de todos los elementos que componen la ventana de la interfaz gráfica.

Esta compuesta de la siguiente forma como se muestra en la figura (15)

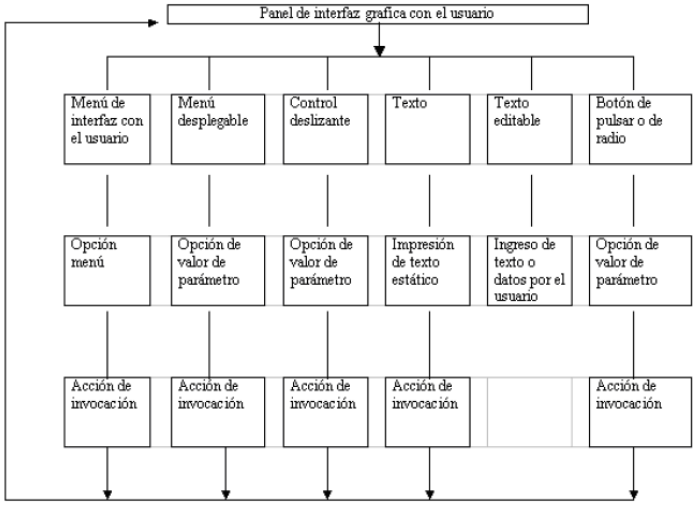


Figura 14: Flujo de operación de GUI

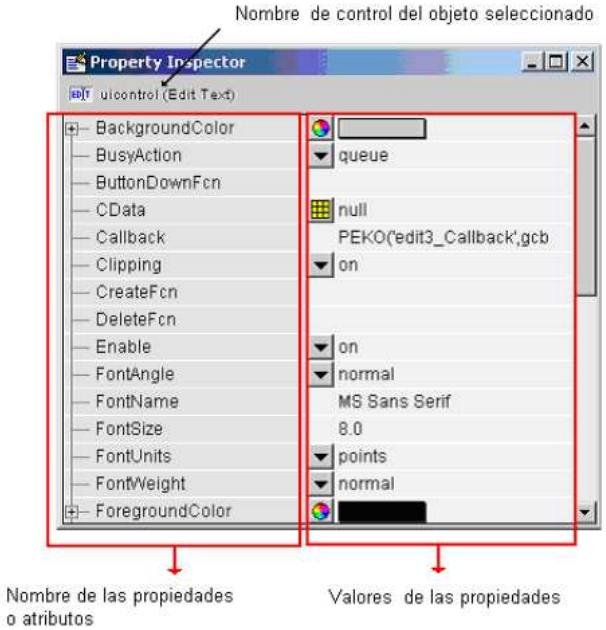


Figura 15: Elementos del Property Inspector

4. Interfaz Gráfica "MathNonParametrics"

4.1. Introducción

Una vez que se ha demostrado la multitud de razones por las cuáles los modelos no paramétricos son útiles en muchos casos en los que los modelos paramétricos no lo son, lo más deseable sería poder implementarlos de forma rápida y sencilla a partir de un conjunto de datos. Por ello, se ha creado la interfaz gráfica denominada MathNonParametrics que permite al usuario aplicar modelos de Regresión Local Polinómica (RLP), una de las técnicas no paramétricas de uso más extendido. La interfaz ha sido desarrollada a partir del programa Guide de Matlab (presentado previamente en este proyecto).

El objetivo es que el usuario pueda usar la interfaz para implementar los modelos de RLP, de manera sencilla y rápida, sin necesidad de un conocimiento profundo de esta técnica. En la interfaz, el usuario tendrá la posibilidad de variar los distintos "parámetros"del modelo, con el fin de conseguir el mejor ajuste posible. Entre estos parámetros, se encuentran, por ejemplo: el tipo de función kernel, ancho de banda, uso de validación cruzada, etc. Además el usuario tendrá la posibilidad de poder analizar los resultados y la bondad del ajuste mediante resultados numéricos y gráficos. Y también, tendrá la posibilidad de grabar los resultados más importantes, como, estimaciones de las variables salida, errores cometidos, etc.

Aunque la función principal de la interfaz son los modelos de Regresión Local Polinómica, se ha desarrollado también la opción de ejecutar Regresión Lineal Múltiple (modelo paramétrico), con el fin de que el usuario pueda comparar y juzgar por si mismo qué tipo de modelo se ajusta mejor a los datos. Hay que tener en cuenta, que un modelo paramétrico que permite obtener buenos resultados será una mejor opción que un modelo no paramétrico.

El aspecto que presenta la interfaz gráfica es el que muestra la Figura (16).

4.2. Información que debe conocer el usuario antes de iniciar la interfaz gráfica

4.2.1. Ubicación de los archivos y versiones de MATLAB

Con la documentación del presente Proyecto Fin Carrera se entrega de forma adjunta un Cd de datos que contiene todo lo necesario para que cualquier usuario pueda ejecutar la interfaz gráfica "MathNonParametrics". Lo único que necesitará será el programa informático MATLAB en alguna de las siguientes versiones:

- R14 (Matlab 7.0)
- R2007a

Figura 16: Pantalla principal

Para versiones más antiguas no se garantiza el correcto funcionamiento del programa, debido a que para la creación de la interfaz gráfica se han utilizado librerías que podrían no estar presentes.

4.2.2. Organización del Cd de datos

El Cd de datos se organiza de la siguiente manera (se muestra la estructura en árbol de la carpeta del Cd):

- Carpeta: *Interfaz MathNonParametrics*

1. Subcarpeta: **Versión Matlab R14 (Matlab 7.0)**
   a) *ArchivosInterfaz*
   b) DirectorioTrabajo
   c) ArchivosAyuda
2. Subcarpeta: **Versión Matlab 2007a**
   a) *ArchivosInterfaz*
   b) DirectorioTrabajo
   c) ArchivosAyuda

Además se incluye un archivo en pdf, denominado InstalacionInterfaz, que contiene la información que ha sido descrita arriba y la que aparece a continuación.

Una vez seleccionada la versión que se va a utilizar, se debe copiar la carpeta **Versión Matlab R14** o **Versión Matlab 2007a** en un directorio seleccionado por el usuario. Por ejemplo, el usuario podría copiar todo en el **escritorio** de su ordenador.

El contenido de las subcarpetas contenidas en la carpeta para ambas versiones es el siguiente:

- *ArchivosInterfaz*: contiene todos los programas que la versión correspondiente necesita para ejecutar la interfaz de forma correcta. Esta subcarpeta NO se modificará por el usuario. Si lo hace se puede correr el riesgo de que el programa no funcione y dé errores inesperados.

- *DirectorioTrabajo*: esta será la subcarpeta que el usuario podrá modificar a su gusto. La función de ésta será la de albergar los datos que se introducirán al programa y los que se obtendrán del mismo. Por defecto se encontrarán los siguientes archivos por si el usuario no dispone de semejantes para ejecutar la interfaz:

  - datos.txt
  - datosval.txt
  - datos2.txt

- *ArchivosAyuda*: en esta subcarpeta se han guardado todos los archivos en formato .pdf que el programa reporta cuando se hace uso de la ayuda. Si el usuario lo desea puede abrir ésta y leer cualquier archivo de ayuda que desee. Se recomienda tener precaución de no borrar ningún archivo contenido en la subcarpeta, debido a que pueden aparecer errores en la ejecución de la interfaz.

4.2.3. Localización de las carpetas desde el Current Directory de MATLAB

Para poder ejecutar la interfaz, el usuario debe seleccionar como directorio principal del MATLAB, el directorio **Versión Matlab R14** o **Versión Matlab 2007a.** Por ejemplo, siguiendo la recomendación de copiar una de estas carpetas en el escritorio de windows, debe aparecer en el Current Directory de MATLAB (como en la figura (17)) las siguietes rutas:

- **Versión R14**: Ç:\Documents and Settings\Administrador\Escritorio\Versión Matlab R14"

- **Versión R2007a**: Ç:\Documents and Settings\Administrador\Escritorio\Versión Matlab R2007a"

4.3. Hitos principales en el uso de la interfaz gráfica

Para utilizar la interfaz gráfica, el usuario debe conocer que de forma general existen las siguientes partes diferenciadas:

- **Pantalla Principal**.
- **Ayuda**.

  Desde la pantalla ayuda se puede acceder a todos los textos de ayuda del programa. Además se ha incluído una pestaña en la parte superior de cada pantalla de la interfaz, que abre la ayuda correspondiente a esa pantalla.

- **Carga de Datos**.

  En esta sección, el usuario introducirá en el programa todos los archivos de datos que después usará a su paso por todas las pantallas de la interfaz

- **Análisis Previo de Datos**.

  Si el usuario no está familiarizado con los datos que ha introducido al programa, se puede llevar a cabo un análisis estadístico de los mismos, de tal forma que el usuario conozca, previamente, las características descriptivas más importantes de la muestra, como por ejemplo, tipo de distribución, si existen datos atípicos, etc.

- **Transformaciones de los datos**.

  En multitud de aplicaciones numéricas, es muy común aplicar una transformación a los datos de tal forma que se pueda trabajar con ellos de una manera mucho más cómoda.

- **Regresión Lineal Múltiple**.

  Esta opción permite ajustar los datos a un modelo paramétrico sencillo. Si este modelo permite un buen ajuste, el usuario podrá trabajar con modelos muchos más cómodos y no tener que hacer uso de modelos mucho más complicados, como el de RLP.

- **Ajuste de parámetros y análisis del modelo (Regresión Local Polinómica)**.

  El uso de RLP requiere que se predefinan un conjunto de parámetros que servirán para ejecutar el modelo a un conjunto de datos. De esta forma, el usuario podrá ajustar lo mejor posible los datos. Una vez que el usuario ha especificado y ejecutado el modelo a un conjunto de datos (datos de entrenamiento y validación) deberá comprobar la bondad de ajuste del modelo, por medio de gráficos y resultados numéricos que el programa reportará.

- **Aplicar Regresión Local Polinómica a nuevos datos**.

  Una vez que el usuario ha ajustado el modelo con los datos de entrenamiento y validación, lo podrá ejecutar a un nuevo conjunto de datos de entrada para la obtención de unos datos estimados de salida.

- **Guardar datos emitidos por el programa**.

  Una vez que el programa se ejecute, el usuario podrá guardar los datos que se han emitido.

4.4. Pantalla Principal

Para ejecutar la interfaz gráfica, el usuario deberá realizar los siguientes pasos:

1. Abrir el programa MATLAB.

2. Seleccionar el directorio principal de Matlab, como se indicó antes. Para que el usuario conozca que está en la carpeta correcta, en la barra superior de Matlab debe aparecer, por ejemplo, la secuencia que se muestra en la figura (17).

Figura 17: Directorio Principal de Matlab para ejecutar la Interfaz.

3. Escribir el nombre de la interfaz en la línea de comandos (pantalla principal de Matlab), como se muestra en la figura (18). Se recomienda que el usuario preste atención al cambio de letra mayúscula-minúscula, debido a que Matlab tiene en cuenta este aspecto.

Figura 18: Ejecución de la interfaz gráfica

Una vez que el usuario accede a la pantalla inicial de la interfaz, figura (19), debe elegir entre realizar las opciones que se muestran.

- **Comenzar programa**: A partir de este botón, el usuario podrá continuar a la pantalla de carga de datos.

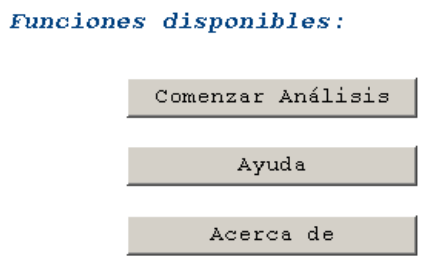


Figura 19: Funciones disponibles en la pantalla principal.

- **Ayuda**: se abrirá una pantalla donde el usuario podrá consultar cualquier duda que tenga respecto a las pantallas de la interfaz y su funcionamiento.
- **Acerca De**: se mostrará una breve información del programa, tal como quién es su autor, fecha, etc.

Nota importante: Cada vez que el usuario se sitúe en la Pantalla principal, se reiniciarán y se fijarán por defecto todos los valores y parámetros del programa.

4.5. Ayuda

A través de esta pantalla, el usuario podrá consultar información acerca del funcionamiento de la interfaz. El aspecto general que presenta se muestra en la figura (20).

Para consultar la información que alberga el menú ayuda, lo único que tiene que hacer el usuario es pulsar el botón correspondiente y esperar que aparezca la información en un documento con formato .pdf (Acrobat Reader)

A la pantalla mostrada en la figura (20) sólo se podrá acceder mediante la pantalla principal (tenga cuidado el usuario pues se reinician los parámetros del programa). Si ha iniciado el programa y desea consultar la ayuda, debe pulsar el menú que aparece en la zona superior de la pantalla, como se muestra en la figura (21).

4.6. Carga de Datos

4.6.1. Aspectos Generales

En esta pantalla se llevará a cabo la carga de datos al programa. Estos datos serán modificados y tratados y serán el punto de partida de los nuevos datos que el programa proporcionará.

Cuando el usuario se sitúe en este punto, deberá introducir como mínimo un archivo de datos para que el programa pueda funcionar. Si por descuido no se introduce ningún archivo

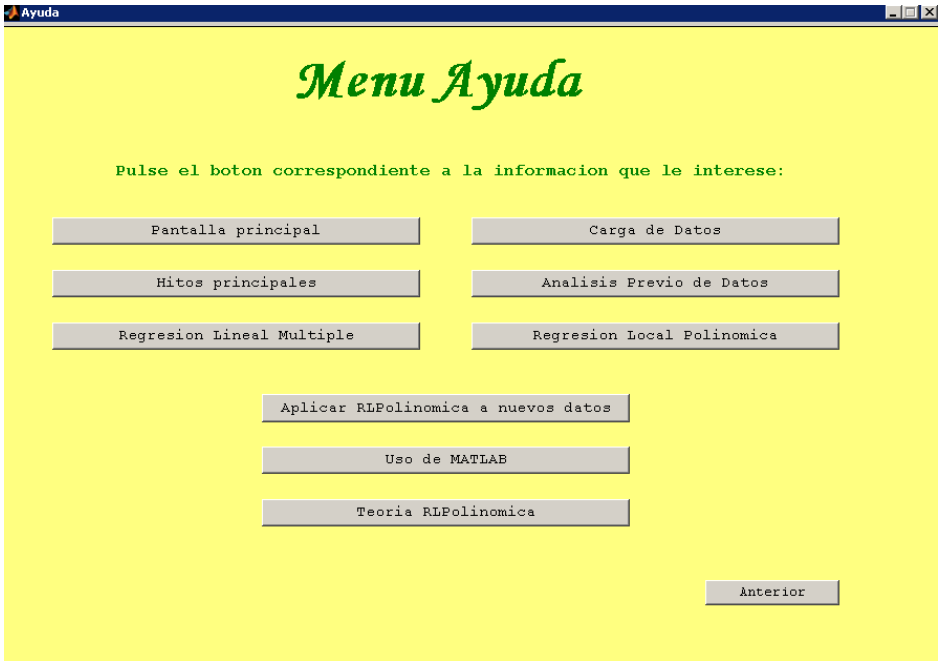


Figura 20: Menú Ayuda

Figura 21: Consultar ayuda una vez iniciado el programa.

de datos, el programa dará un mensaje de aviso como el que se muestra en la figura (22).

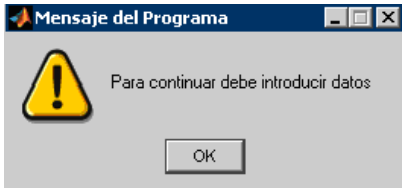


Figura 22: Aviso de carga de datos.

Un aspecto muy importante es que el usuario debe guardar el archivo de datos en la carpeta "**DirectorioTrabajo**", que es la carpeta que la interfaz usa para ir grabando también todos los resultados, en ficheros txt. Además, el archivo debe estar en formato .txt. Debido a que multitud de usuarios seguramente posean datos contenidos en otro tipo de programas que no son hojas .txt, en la sección de Otras peculiaridades, se mostrará un procedimiento para pasar los datos en formato cualquiera a .txt con el formato que la interfaz requiere. El separador de decimales debe ser "**.** ".

En el momento que el usuario tenga los datos cargados, el programa le permitirá continuar sin problemas.

4.6.2. Explicación de campos en el panel de carga de datos

El aspecto general que presenta la pantalla de datos es el mostrado en la figura (23):

Figura 23: Aspecto general de la pantalla

La pantalla de Carga de Datos se divide en las siguientes secciones:

1. **Zona donde se introduce el nombre del archivo** seguido de .txt y los botones que activan la carga de datos. El programa permite almacenar tres archivos diferentes que podrán ser utilizados en cualquier momento a lo largo del uso del programa.

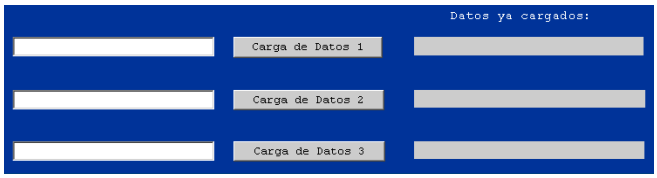


Figura 24: Zona almacenamiento de datos.

2. **Botones de salida de la pantalla.** Estos botones permitirán abandonar la pantalla de Carga de datos de forma que se siga de forma correcta en el programa. El funcionamiento de ellos es conducir a las siguientes pantallas. Estos botones son:

- Análisis previo de datos
- Aplicar Regresión Local Polinómica

Además de todo lo que se ha mencionado, el programa posee un botón que se sitúa en la zona de menús donde el usuario podrá imprimir en cualquier momento la pantalla que está visualizando.

Figura 25: Botones de Imprimir y Ayuda, repectivamente.

4.6.3. Pasos a seguir para realizar la carga de datos.

En este apartado se va a mostrar un ejemplo de proceso a seguir para llevar a cabo la correcta carga de datos al programa para poder usarlos después de forma cómoda.

Paso 1 Antes de comenzar, el usuario deberá contener los datos en un archivo .txt. Además, el archivo deberá estar guardado en la carpeta "DirectorioTrabajo". El formato que deberá contener el archivo .txt deberá ser como el que muestra la figura (26):

Notar, que si se desea poner nombre a las columnas de los datos, deben ir precedidas de un símbolo de porcentaje, "%"(marcado en la figura (26)) para que el programa no dé error.

%A	B	n	FRICTION	Fx	Fy	s11
300	900	0.2	0.7	290	175	380	550
300	900	0.3	0.7	358	238	669	796
300	900	0.3	0.6	324	191	777	830
300	900	0.3	0.5	300	162	786	790
300	900	0.4	0.5	360	189	1040	1003
300	900	0.4	0.4	322	144	1045	984
300	900	0.5	0.4	371	161	1255	1151
300	900	0.5	0.5	385	195	1230	1150
300	800	0.3	0.4	256	118	755	683
300	800	0.3	0.5	286	152	738	750
300	800	0.4	0.6	351	212	933	946
300	700	0.4	0.5	360	166	1030	1030
300	700	0.4	0.6	331	198	846	874
300	500	0.5	0.5	300	155	970	970
300	600	0.4	0.5	301	157	846	848
300	600	0.4	0.4	288	127	831	826
300	600	0.5	0.4	308	132	1018	971
400	900	0.4	0.4	306	141	980	951
400	900	0.4	0.5	342	177	987	957
400	900	0.4	0.6	378	226	909	940
400	900	0.4	0.7	400	250	800	660
400	900	0.508	0.4	364	161	1125	1095
400	900	0.508	0.6	396	240	1140	1097
400	900	0.6	0.4	394	171	1340	1268
400	900	0.6	0.5	470	220	1410	1365
400	800	0.6	0.5	392	204	1210	1212

Figura 26: Ejemplo formato archivo .txt

Si por alguna razón el usuario contiene los datos en otro tipo de formato, en la sección 'Otras peculiaridades' se le indicará como puede realizar el cambio de un formato a otro de forma sencilla.

Paso 2 El siguiente paso que deberá realizar el usuario es el de escribir el nombre del archivo txt en el editor que se desea que almacene los datos. Si por ejemplo, el usuario desea cargar el archivo "datos", deberá escribir: datos.txt.

Lo siguiente que deberá realizar el usuario en este paso será pulsar el botón que indica carga de datos. Si por ejemplo el usuario desea cargarlos en la primera ranura de datos, deberá escribir en el editor 1 el nombre del archivo para después pulsar el botón "Carga de Datos 1".

En el momento que el programa detecte que se ha realizado todo correctamente, se mostrará el siguiente mensaje de aviso que indica la figura (28).

Nota importante: como criterio a seguir, se cargarán siempre los datos de entrenamiento en Carga de datos 1 (ranura de datos 1) y los de validación y estimación en las ranuras de datos 2 y 3.

Se han cargado unos ficheros de datos, para que el usuario pueda utilizarlos como ejemplo. Estos ficheros tienen los siguientes nombres:

- Editor de la ranura de datos 1: "**datos.txt**"(datos para entrenamiento).
- Editor de la ranura de datos 2: "**datosval.txt**"(datos para validación).
- Editor de la ranura de datos 3: "**datos2.txt**"(datos para estimar las variables salidas

Figura 27: Indicación de cómo cargar los datos

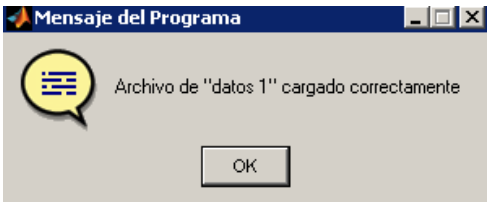


Figura 28: Mensaje que indica que todo se ha cargado correctamente.

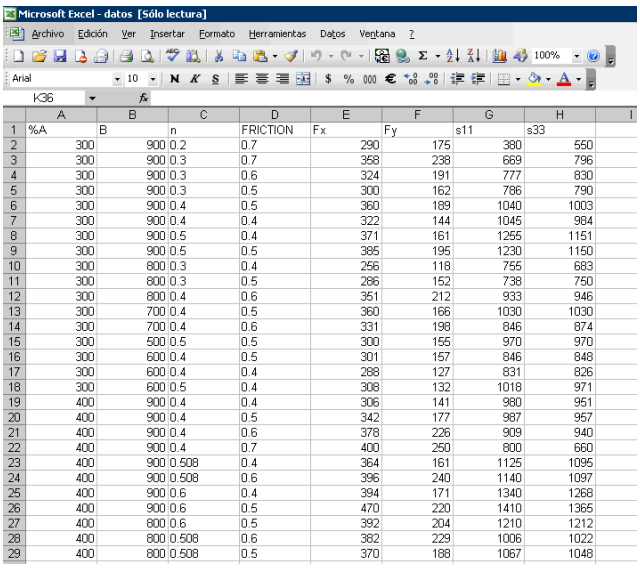
dados unos valores de entrada).

Paso 3 En el momento que se muestre la pantalla indicada en la Figura (28), el usuario ya podrá continuar su paso por el programa. En este momento deberá decidir si continua por una de las tres siguientes pantallas:

- Análisis previo de datos
- Aplicar Regresión Local Polinómica

4.6.4. Otras peculiaridades

Como peculiaridad importante, se mostrará el procedimiento para pasar los datos que posee el usuario almacenados en un archivo de hoja Excel a otro en el formato .txt que el programa requiere.



	A	B	C	D	E	F	G	H	I
1	%A	B	n	FRICTION	Fx	Fy	s11	s33	
2	300	900	0.2	0.7	290	175	380	550	
3	300	900	0.3	0.7	358	238	669	796	
4	300	900	0.3	0.6	324	191	777	830	
5	300	900	0.3	0.5	300	162	786	790	
6	300	900	0.4	0.5	360	189	1040	1003	
7	300	900	0.4	0.4	322	144	1045	984	
8	300	900	0.5	0.4	371	161	1255	1151	
9	300	900	0.5	0.5	385	195	1230	1150	
10	300	800	0.3	0.4	256	118	755	683	
11	300	800	0.3	0.5	286	152	738	750	
12	300	800	0.4	0.6	351	212	933	946	
13	300	700	0.4	0.5	360	166	1030	1030	
14	300	700	0.4	0.6	331	198	846	874	
15	300	500	0.5	0.5	300	155	970	970	
16	300	600	0.4	0.5	301	157	846	848	
17	300	600	0.4	0.4	288	127	831	826	
18	300	600	0.5	0.4	308	132	1018	971	
19	400	900	0.4	0.4	306	141	980	951	
20	400	900	0.4	0.5	342	177	987	957	
21	400	900	0.4	0.6	378	226	909	940	
22	400	900	0.4	0.7	400	250	800	660	
23	400	900	0.508	0.4	364	161	1125	1095	
24	400	900	0.508	0.6	396	240	1140	1097	
25	400	900	0.6	0.4	394	171	1340	1268	
26	400	900	0.6	0.5	470	220	1410	1365	
27	400	800	0.6	0.5	392	204	1210	1212	
28	400	800	0.508	0.6	382	229	1006	1022	
29	400	800	0.508	0.5	370	188	1067	1048	

Figura 29: Hoja de datos de Microsoft Excel

Partiendo que el usuario se sitúa en una hoja Excel como se muestra en la Figura (29), el usuario lo único que deberá hacer es pulsar: Archivo, Guardar Como, Texto (delimitado por tabulaciones) y pulsar.Aceptar en la pantalla que le sigue. Con este procedimiento, el usuario tendrá garantizado que el programa aceptará sus datos.

4.7. Análisis Previo de Datos (Análisis Descriptivo)

4.7.1. Aspectos Generales

A esta pantalla el usuario puede acceder a través de un botón situado en la pantalla de Carga de Datos en el lado derecho, como muestra la figura (23).

El objetivo que persigue esta pantalla es que el usuario conozca algunas **características univariantes** de los datos que ha cargado al programa. Se proporcionan las siguientes herramientas:

- Número de variables y datos por variable
- Visor de datos que contiene cada variable
- Media
- Varianza
- Coeficiente de asimetría
- Coeficiente de curtosis
- Histogramas
- Diagramas Box Plot

A partir de esta información, el usuario puede tener una idea más clara sobre el conjunto de datos que quiere modelizar. Por ejemplo, si las variables no son normales, si no que tienen una distribución asimétrica, quizás sería conveniente realizar una transformación previa, como raíz cuadrada. Otra utilidad de este análisis es la posibilidad de localizar observaciones atípicas, mediante los boxplots, o histogramas. Sin embargo, hay que tener en cuenta, que este análisis es univariante y por lo tanto debe ser tratado con mucho cuidado.

4.7.2. Explicación de campos

El aspecto general que presenta la pantalla de datos es el siguiente:
La pantalla de Análisis de Datos se divide en las siguientes secciones:

1. **Zona donde se introduce el lugar donde están almacenados los datos que se desea analizar**. Por defecto se muestra la ranura de datos 1.
2. Zona donde se indica el número de variables que existen, así como el número de datos que contiene cada variable.

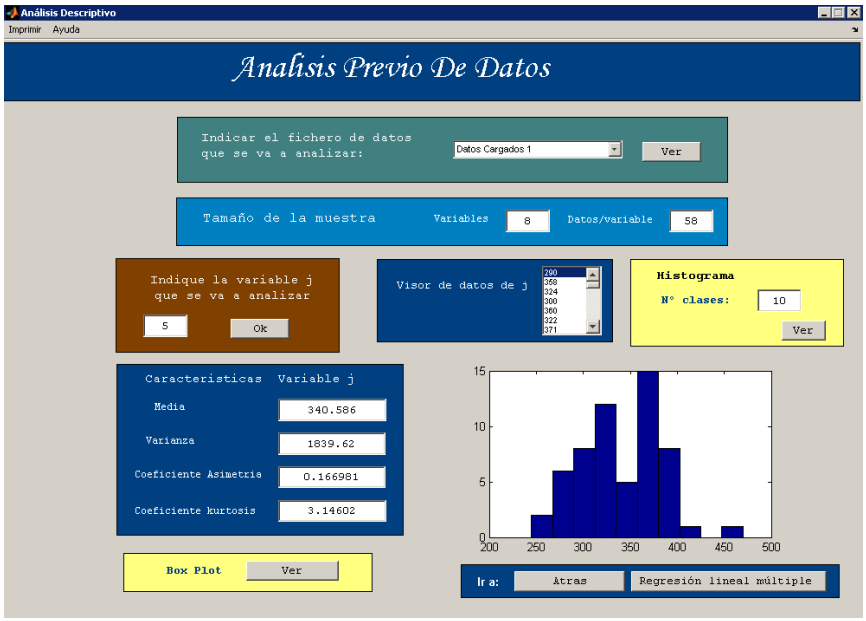


Figura 30: Aspecto general pantalla Análisis de Datos

Figura 31: Información sobre datos a analizar

3. **Zona de selección de la variable contenida en la muestra que se desea analizar**. Cada vez que se modifique en el editor el número de variable, el usuario debe pulsar el botón Ok para hacer efectiva el cambio.

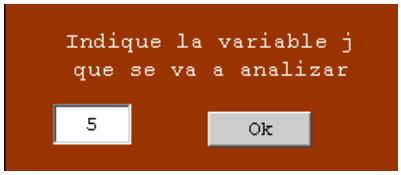


Figura 32: Número de variable a analizar

4. **Zona de análisis estadístico de los datos**. Esta zona consta únicamente de información estadística en formato numérico. A partir de ella, el usuario podrá tomar decisiones sus primeras decisiones sobre los análisis que se realizarán posteriormente.

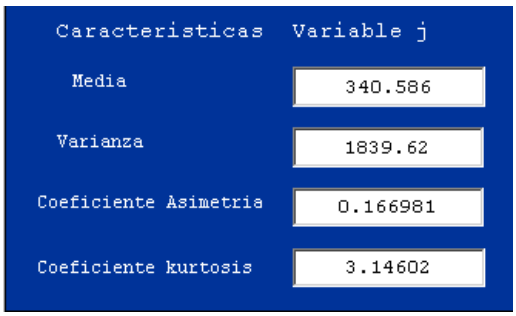


Figura 33: Información estadística de los datos

5. **Zona de gráficos**. Esta zona puede proporcionar dos tipos diferentes de gráficos: histograma y diagramas box plot. Para usarlos, el usuario tiene que haber elegido una variable previamente, como se ha indicado en la figura (32).

4.7.3. Pasos a seguir para el análisis previo de datos

En este apartado se va a mostrar un ejemplo de proceso a seguir para llevar a cabo el correcto análisis de datos cargados al programa para poder tomar decisiones previas que ayuden a mejorar el uso del programa.

Paso 1 El primer paso que el usuario debe llevar a cabo será el de elegir en qué ranura están los datos que quiere analizar, como muestra la figura (36).

Una vez que se haya elegido el tipo de ranura el usuario debe pulsar el botón Ver para que se carguen los datos de forma correcta en la pantalla de análisis.

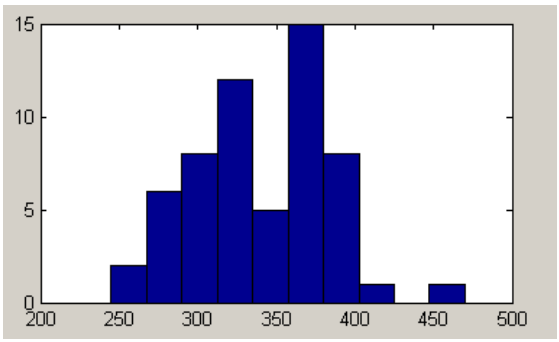


Figura 34: Histograma

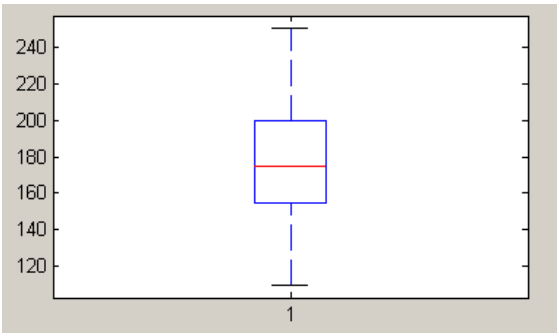


Figura 35: Diagrama de caja

Figura 36: Selección de la ranura que contiene los datos.

Paso 2 Cuando se hayan cargado los datos que se desean analizar, como se ha mostrado en la figura (36), el siguiente paso que se debe llevar a cabo es el de elegir el número de variable contenida en los datos que se desea analizar. Por defecto aparecerá el número 1. Para hacer efectiva la selección, se debe pulsar el botón Ok, como se ha mostrado en la figura (32).

En este momento, se cargará automáticamente la información estadística contenida en los datos de la variable pero no se cargará ningún gráfico de análisis. Para ello, el usuario deberá pulsar uno de los botones, es decir, dependiendo si desea Histograma o Diagrama Box Plot, como se mostrado en las figuras (34) y (35).

Una vez realizado todo lo descrito en el paso 2, el usuario si lo desea podrá continuar a la siguiente para ejecutar el modelo de Regresión Lineal Múltiple o regresar a la pantalla de Carga de Datos para continuar con el modelo de Regresión Local Polinómica.

4.8. Regresión Lineal Múltiple

4.8.1. Aspectos Generales

En esta pantalla se puede ajustar los datos usando un modelo paramétrico lineal: Regresión Lineal Múltiple. Puede obtenerse un buen ajuste, o por el contrario, puede ser que este modelo no sea el adecuado, por lo que sería recomendable pasar al modelo de Regresión Local Polinómica (RLP).

Cuando el usuario se sitúe en este punto, deberá decidir qué variables de las que contiene el archivo de entrenamiento desea introducir como variables de entrada y cuáles como variables de salida. El programa por defecto le mostrará unas variables de entrada/salida.

Un aspecto importante que el usuario debe tener en cuenta es la ubicación de los datos de entrenamiento. Como criterio, se aconseja que se almacenen en la ranura de datos número 1.

Figura 37: Ubicación de los datos de entrenamiento

Otro aspecto importante es que cada vez que el usuario entre de nuevo en esta pantalla, deberá pulsar el botón **Run**, de lo contrario el programa avisará de que lo debe hacer para continuar.

En el momento que el usuario haya pulsado del botón de ejecución, el programa le permitirá continuar sin problemas.

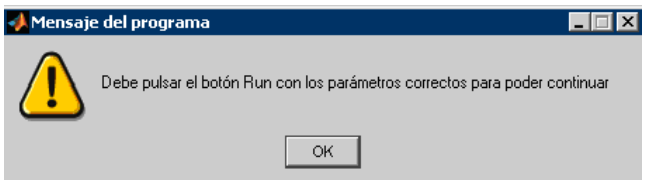


Figura 38: Mensaje de aviso del programa

4.8.2. Explicación de Campos

El aspecto general que presenta la pantalla de datos es el mostrado en la figura (39).

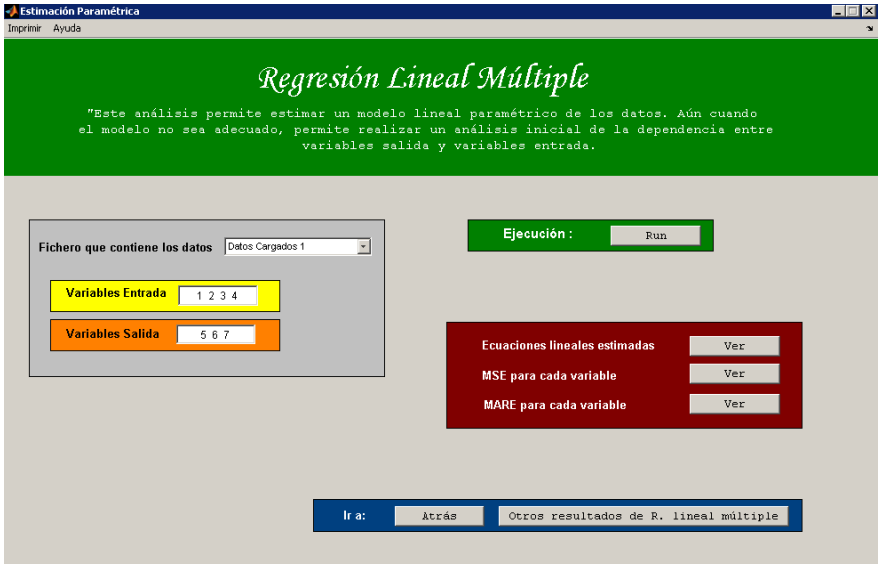


Figura 39: Aspecto general de la pantalla.

La pantalla de Regresión Lineal Múltiple se divide en las siguientes secciones:

1. **Zona de selección de datos a utilizar y variables entrada/salida**. Como se indica en la figura (40).

2. **Zona de ejecución**. Una vez que el usuario haya seleccionado los datos a utilizar y las variables entrada/salida del modelo, deberá ejecutar para poder obtener resultados. Se indica en la figura (41).

3. **Primera zona de datos emitidos por el programa**. Se trata que el programa reporte los datos numéricos que ha obtenido del modelo, como son las ecuaciones lineales estimadas (figura (43)), MSE (error cuadrático medio) (figura (44)) para cada variable y MARE (%) (error absoluto relativo medio) (figura (45)) para cada variable. Esta zona

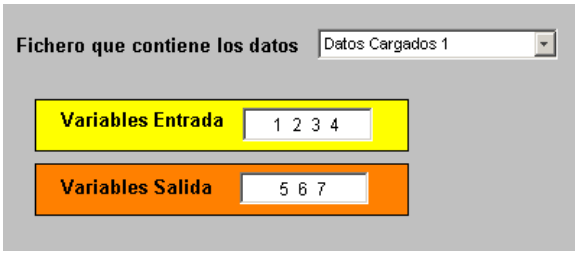


Figura 40: Selección de los datos a utilizar

Figura 41: Botón ejecución

se indica en la figura (42). Estos resultados se graban en la carpeta "DirectorioTrabajo", en unos ficheros txt. Para que se generen es necesario que el usuario pulse todos los botones de las figuras mencionadas en este tercer apartado. Los nombres de los ficheros .txt que se guardan son los siguientes:

- Ecuaciónes lineales estimadas:
  - Matriz Beta: "Beta_parametrica.txt"
  - Matriz X: "Matriz_X_parametrica.txt"
  - Matriz Y estimada: "Matriz_Yestimada_parametrica.txt"
- MSE: "MSE_parametrica.txt"
- MARE: "MARE_parametrica.txt"

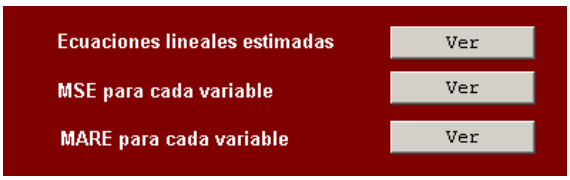


Figura 42: Datos emitidos por el modelo

En esta pantalla se ha creado para que el usuario ejecute el modelo. Para ver los resultados de forma gráfica que nos ha proporcionado el programa, el usuario deberá pulsar en el botón .ºtros resultados de la R.Lineal Múltiple", como se indica en la figura (46).

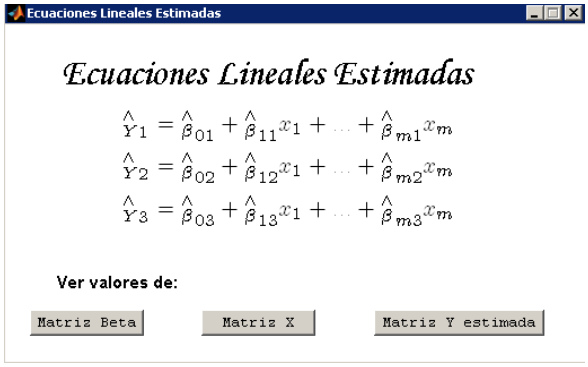


Figura 43: Ecuaciones Lineales Estimadas en Regresión Lineal Múltiple.

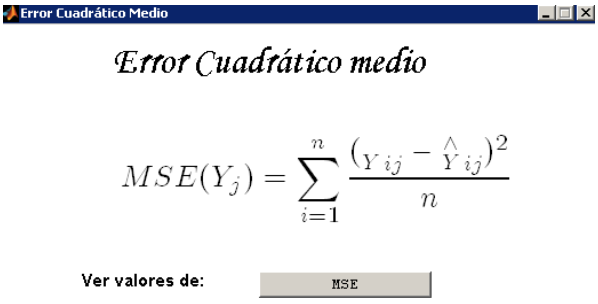


Figura 44: Error cuadrático medio

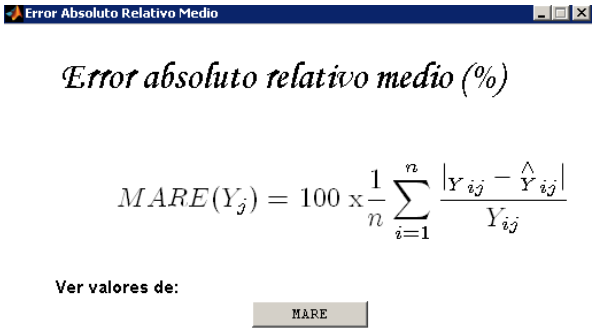


Figura 45: Error absoluto relativo medio (%)

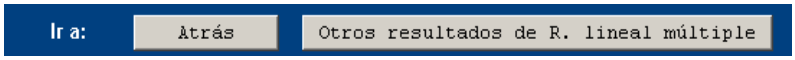


Figura 46: Ir a resultados de la Regresión Lineal Múltiple.

4.8.3. Resultados Regresión Lineal Múltiple

El aspecto general que muestra esta pantalla, es el que se indica en la figura (47).

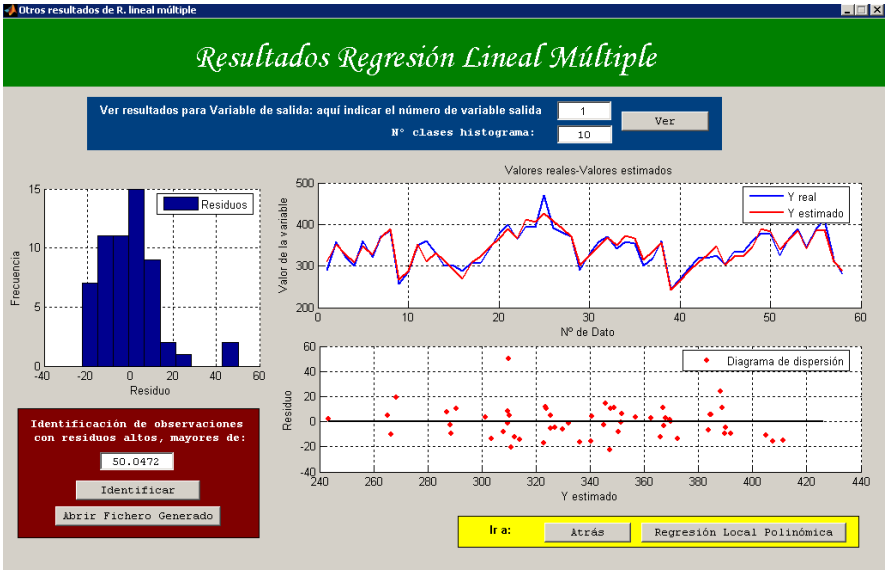


Figura 47: Aspecto general de la pantalla de resultados de la Regresión Lineal Múltiple.

La pantalla de resultados de Regresión Lineal Múltiple se divide en las siguientes secciones:

1. **Selección del número de variable de salida y del número de clases del histograma**, como se muestra en la figura (48).

   Si el usuario en la pantalla previa (figura (39)) seleccionó, por ejemplo, 3 variables de salida: variables 5, 6 y 7 del fichero; entonces en esta pantalla deberá poner un 1 para la variable número 5, un 2 para la variable número 6 y un 3 para la variable número 7, y así sucesivamente si se seleccionaron más variables de salida.

   El campo de número de clases del histograma hace referencia al número de grupos en los que el usuario quiere dividir los datos.

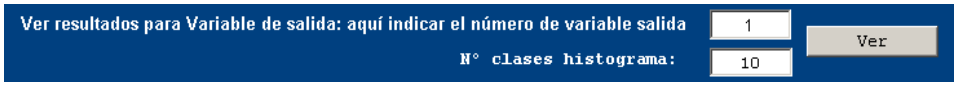


Figura 48: Selección del número de variable de salida.

2. **Resultados gráficos**. En esta zona se agrupan los tres gráficos que se muestran en la pantalla, y que corresponden a:

   - *Histograma de residuos* (figura (49)). Este histograma permite analizar la distribución de los residuos. Si los residuos siguen una distribución normal, podemos pensar en un

buen ajuste de los datos. Si presenta una distribución asimétrica, podría pensarse que el modelo no es el más adecuado, por ejemplo, podrían existir residuos muy altos para ciertos valores de entrada.

- *Valores reales - valores estimados* (figura (50)). Este diagrama, permite comparar los valores de salida reales con los estimados. Si existe mucha diferencia entre ellos, el modelo no será bueno. Por ejemplo, se podría observar que los valores estimados son siempre menores (o mayores) que los reales, o que suele dar una mala estimación cuando el valor de salida es alto, etc.

- *Diagrama de dispersión Residuos vs Yestimada* (figura (51)). Este gráfico es muy útil para medir la bondad de ajuste del modelo. Si el modelo es bueno, los puntos de este gráfico deberían estar distribuidos de forma aleatoria alrededor del cero. Si no es así, se puede pensar en un mal ajuste. Por ejemplo, si los residuos presentan un comportamiento no aleatorio, más valores positivos que negativos, entonces el modelo está segado. Si se observa que la varianza va aumentando, o disminuyendo, entonces se detecta homocedasticidad, que deberá ser corregida. Si los residuos describen una curva, indicaría que hay una relación cuadrática que se ha tenido en cuenta. Si se observan residuos muy altos, en valor absoluto, podría tratarse de observaciones atípicas.

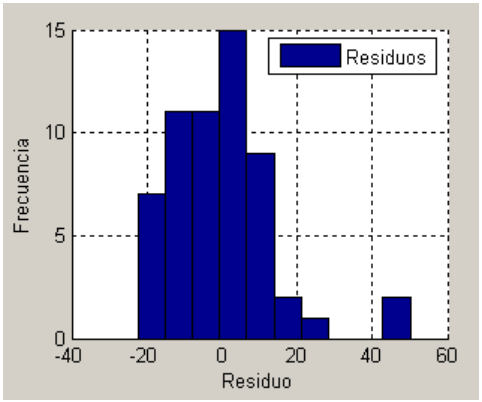


Figura 49: Histograma residuos

3 **Zona de identificación de residuos más altos que un valor predefinido**, figura (52).Este análisis está relacionado con el diagrama de dispersión Residuos vs Yestimada, de la sección anterior. Como se mencionó antes, valores de residuos muy altos, en valor absoluto, pueden deberse a observaciones atípicas. Esta opción permite identificar estas observaciones para que puedan ser analizadas con más cuidado. En esta zona el usuario podrá dar un valor máximo de residuo, en valor absoluto, que cree oportuno a partir

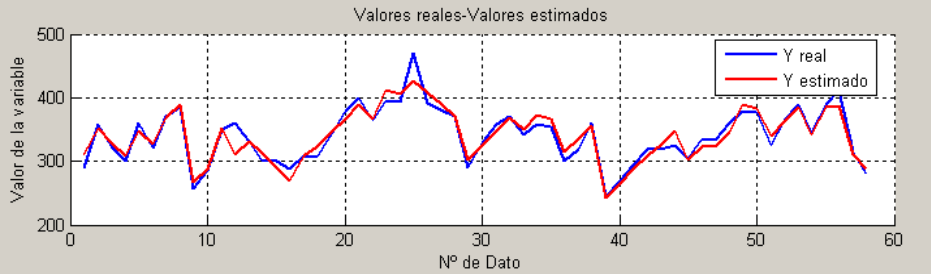


Figura 50: Valores reales - Valores estimados

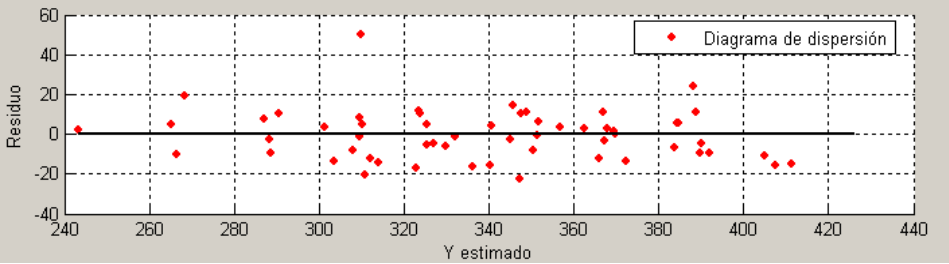


Figura 51: Diagrama de dispersión Residuos - Y estimado

de una vista a la figura (51). En el momento que el usuario dé un valor máximo, podrá pulsar el botón "Identificar". En este momento se generará un fichero que se guardará en la carpeta "DirectorioTrabajo", con el nombre "MatrizResiduosAltos.txt". Si el usuario desea abrir el fichero generado, podrá hacerlo sin problemas desde la pantalla.

La estructura del archivo que se genera ("MatrizResiduosAltos.txt") es la siguiente:

- Primera columna: número de fila que produce el residuo a estudio.
- Antepenúltima columna: valor real de la variable
- Penúltima columna: valor estimado de la variable
- Última columna: residuo que produce la variable estimada
- Demás columnas: valores de las variables de entrada

4.8.4. Pasos a seguir

En este apartado se va a mostrar un ejemplo de proceso a seguir para llevar a cabo uso de los modelos de regresión lineal múltiple. Además el programa ayudará al usuario a comprobar si éste tipo de modelos es el adecuado para los datos que han sido cargados.

Paso 1 El primer paso a llevar a cabo será el de comprobar en qué ranura de almacenamiento de datos están cargados los datos de entrenamiento. Este proceso es sencillo y basta con

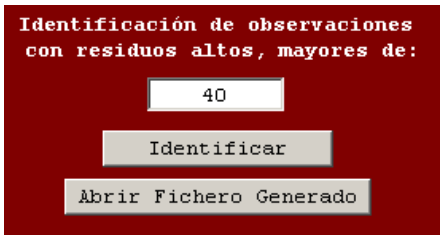


Figura 52: Identificación de residuos mayores de un valor predefinido por el usuario.

regresar a la pantalla de Carga de Datos (23) y comprobarlo si el usuario duda. Además el usuario debe conocer cuántas variables están contenidas en sus datos de entrenamiento. Para conocer este dato puede ir a la pantalla de Análisis de Datos (30).

Cuando el usuario conozca la ubicación de los datos de entrenamiento y cuántas variables están contenidas en ellos, el siguiente paso será seleccionar en la pantalla la ranura de los datos, tal y como se muestra en la figura (40). Además de todo lo descrito, el usuario deberá especificar qué variables de entrada/salida desea introducir al modelo de regresión lineal múltiple.

Una vez que todo lo previo está correctamente realizado, el usuario deberá pulsar el botón Run como se muestra en la figura (41). En ese momento, deberá aparecer un mensaje que comunique que todo ha funcionado correctamente, como muestra la figura (53).

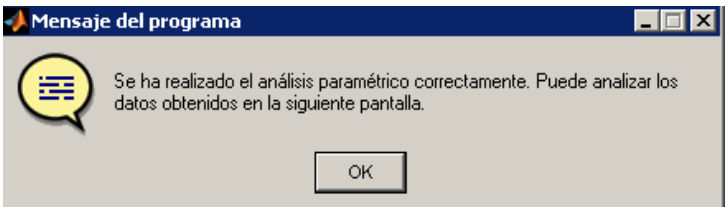


Figura 53: Verificación de que el paso 1 se ha realizado correctamente.

Paso 2 Cuando se haya mostrado por pantalla la figura (53), el usuario podrá realizar un análisis de los datos proporcionados por el modelo paramétrico en la pantalla que muestra la figura (47). Estos análisis se basan en la observación de los gráficos mostrados en las figuras (49), (50) y (51). El usuario podrá actualizar los gráficos cambiando el número de la variable de salida como se muestra en la figura (48), y pulsando el botón "Ver".

Paso 3 El tercer paso consistirá en razonar por parte del usuario si el modelo se ha ajustado bien a los datos que fueron cargados. Para ello, puede hacer uso de la zona de identificación de residuos (figura (52)).

Una vez finalizado el paso 3, el usuario si lo desea podrá continuar con el programa haciendo uso de los modelos de Regresión Local Polinómica, pulsando el botón correspondiente como se indica en la figura (54).

Figura 54: Continuar con el programa con los modelos de Regresión Local Polinómica

4.9. Regresión Local Polinómica

4.9.1. Introducción

Una vez que el usuario ha tenido una primera toma de contacto con el análisis de los datos y la aplicación de los modelos de regresión lineal múltiple, el siguiente paso es la aplicación de los modelos no paramétricos como lo es el modelo de Regresión Local Polinómica. A partir de este punto, lo que resta de las pantallas del programa, se van a dedicar exclusivamente a la utilización de este tipo de modelo.

Hay que destacar que existen dos secciones muy diferenciadas:

1. La **primera** de ellas se basa exclusivamente en lograr que el **usuario encuentre los parámetros óptimos** que consigan que el modelo se adapte a los datos (kernel, ancho de banda, tamaño set validación, etc.). Esta primera parte se podría decir que es un entrenamiento para el modelo. Si el usuario a través de los gráficos/resultados encuentra los parámetros adecuados, puede esperar que el error que cometa el modelo al evaluar nuevos datos sea de orden similar.

   Las pantallas que engloba esta primera sección son las siguientes:

   - Selección de variables de entrada/salida
   - Selección Parámetros Regresión Local Polinómica
   - Ejecución y resultados
   - Grabar resultados

2. La segunda sección correspondería a la **estimación usando nuevos valores de entrada, una vez que los parámetros del modelo están optimizados**. Esta sección sólo funcionará de forma óptima si el usuario ha dedicado el tiempo necesario para calibrar el modelo a los datos que han servido para entrenarlo.

   La pantalla que está contenida en esta sección es:

- Aplicar RLPolinómica a nuevos datos

Primeramente se presentarán todas las pantallas nombradas y luego se indicarán los pasos que hay que llevar a cabo para que el usuario pueda ejecutar el modelo con éxito.

4.9.2. Selección de variables de entrada/salida

Esta pantalla es el punto de partida de los modelo no paramétricos. A partir de ella se seleccionarán todos los parámetros que son necesarios para poder ejecutar este tipo de modelos.El aspecto general que muestra esta pantalla es el que indica la figura (55).

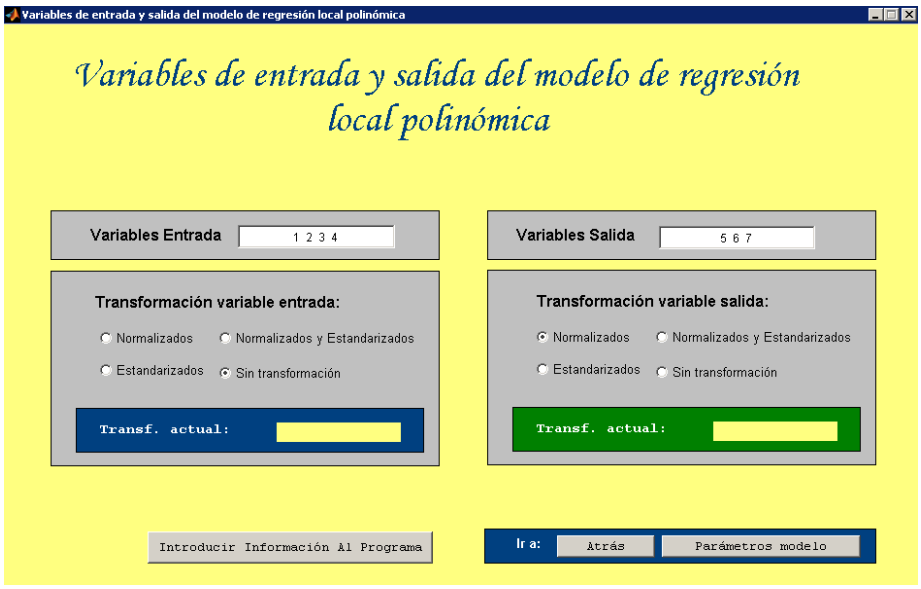


Figura 55: Pantalla de selección de variables entrada / salida

El esquema que sigue la pantalla es el de separa las variables de entrada y salida por separado para aplicarles una transformación diferente que desee el usuario. A la izquierda de la pantalla se deben indicar las variables que se desea que sean variables de entrada para el modelo y que están contenidas en el archivo cargado previamente. Seguidamente se debe indicar una transformación posible para las variables de entrada seleccionadas. De forma análoga, en la zona derecha de la pantalla se deben indicar las variables de salida y una posible transformación. Las transformaciones posibles de la Interfaz son:

1. Normalizados: se dividen los datos de cada columna por el valor máximo contenido en cada una de ellas.

2. Estandarizados: se calculan la media y la desviación típica de los valores de cada columna. Para estandarizarlos, se resta la media a los datos y luego se divide por su desviación típica.

3. Normalizados y estandarizados: se aplica las transformaciones 1 y 2 simultáneamente.

4. Ninguna: en este caso no se aplica ninguna transformación, y por lo tanto los datos presentan los valores de origen.

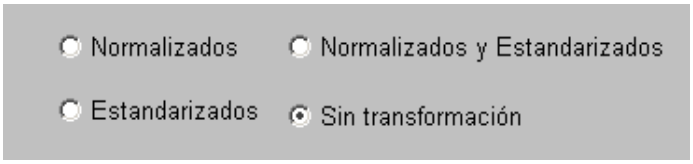


Figura 56: Posibles transformaciones de los datos.

4.9.3. Selección de los parámetros usados en el modelo de Regresión Local Polinómica

Esta pantalla ayudará al usuario a elegir los parámetros que requiere el modelo no paramétrico. Es necesario destacar que la selección óptima de los parámetros del modelo es un proceso iterativo, en el que el usuario, primero da un valor específico a unos determinados parámetros y después analiza los resultados para medir la bondad de ajuste del modelo usando esos parámetros. Para poder realizar esta selección de parámetros, es necesario que se realice una etapa de entrenamiento y validación del modelo. Por tanto, es necesario que el usuario disponga de un conjunto de datos para cada una de estas etapas.

La manera más sencilla de entrenamiento-validación sería, a partir de un conjunto de datos, dividirlo en 2 subconjuntos: uno para usar como datos de entrenamiento y otro como datos de validación. Sin embargo, esto podría no ser lo más adecuado, porque dependerá mucho de la división que se ha hecho. Por ejemplo, en el caso que se disponga de pocos datos, podría suceder, que los datos de entrenamiento elegidos son muy representativos de la población y con pocos errores de medición (sin atípicos, ni valores extremos, etc), y por tanto, se obtendrían buenos resultados en el ajuste del modelo. O, podría suceder lo contrario, que el conjunto de datos de entrenamiento no sea el mejor de la muestra y por tanto, el modelo no daría buenos resultados.

Por esta razón, otra manera de realizar la selección de parámetros del modelo, es usando lo que se llama **Validación cruzada (cross-validation).** Mediante esta técnica, lo que se hace es dividir muchas veces, el conjunto de datos en un subconjunto de entrenamiento y otro de validación, y probar el modelo con dichos subconjuntos (cada vez que se pruebe con unos subconjuntos, se cuenta como una iteración). El error cometido en el ajuste será calculado como el error promedio obtenido al usar todas las divisiones diferentes.

En la Interfaz, se han incluido estas dos opciones: Usar **Cross-validation y No usar**

Cross-validation. Se describirán a continuación:

Validación Cruzada Si se desea ajustar los parámetros del modelo no paramétrico a través de Validación Cruzada, es necesario completar todas las variables que muestra la figura (57).

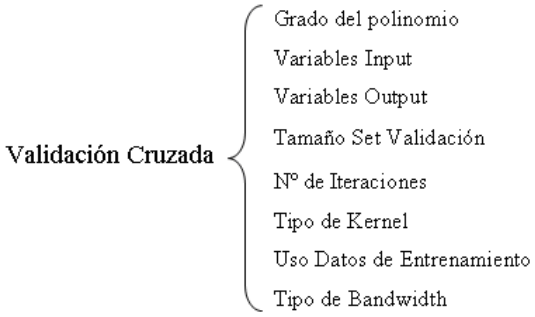


Figura 57: Parámetros necesario en "validación cruzada".

Para medir el error cometido por el modelo en todas las iteraciones hechas, se usará el Error Absoluto Relativo Medio: MARE %, obtenido como promedio de los errores cometidos para todas las variables salida y todas las iteraciones. El cálculo se realiza mediante la siguiente expresión:

$$MARE(\%)(Y_j) = 100\frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_{ij} - \hat{Y}_{ij}\right|}{Y_{ij}}$$

Sin Validación Cruzada Si se desea ajustar los parámetros del modelo no paramétrico a través de Sin Validación Cruzada, es necesario completar todas las variables que muestra la figura (58).

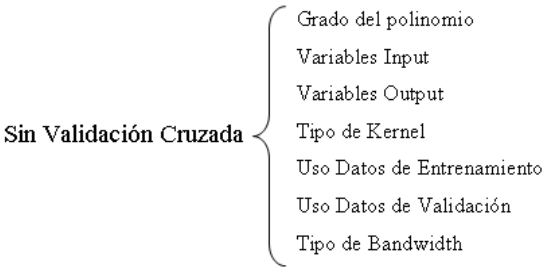


Figura 58: Parámetros necesarios en la opción "sin validación cruzada".

El modelo también se evalúa usando el MARE %, pero en este caso, sólo hay una iteración, por lo tanto, el error se obtiene como el promedio del error cometido para todas las variables salida en esa iteración. Es importante que para usar esta opción, es necesario cargar al

programa un conjunto de entrenamiento y otro de validación (en (23), ambos con la misma estructura (igual número de columnas, en el mismo orden). Si esto no se tiene en cuenta, el programa no funcionará correctamente.

Explicación de Campos El aspecto general que muestra la pantalla de selección de parámetros para el modelo de regresión local polinómica es el que indica la figura (59).

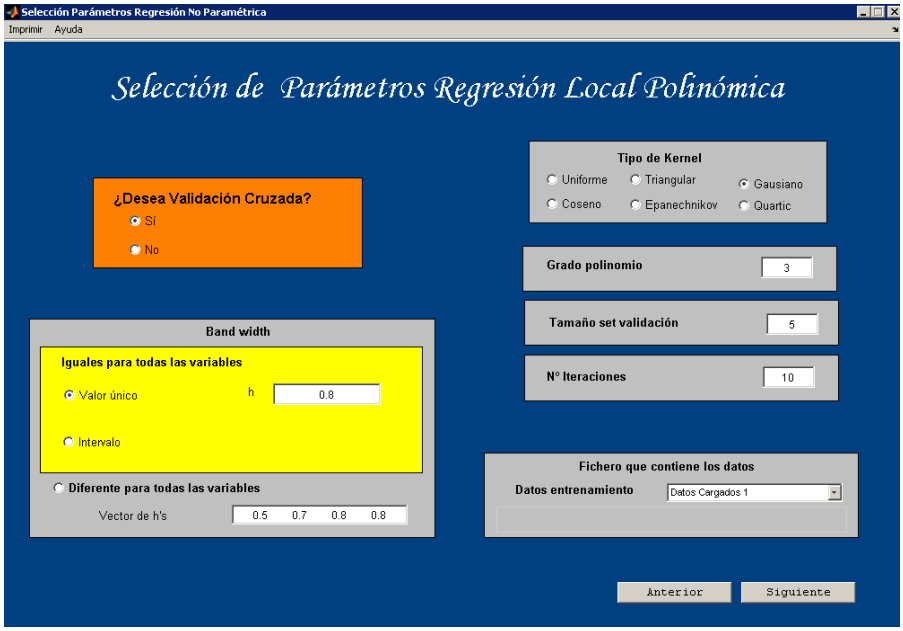


Figura 59: Aspecto general pantalla de selección de parámetros.

La pantalla de Selección Parámetros para el modelo de regresión local polinómica se divide en las siguientes secciones:

1. **Selección de de la forma en la que se definirán los parámetros del modelo**, es decir, con o sin validación cruzada. Por defecto se muestra activa Validación Cruzada Sí. Este campo definirá dos caminos completamente diferentes de selección de parámetros. Se recomienda al usuario ver las figuras (57) y (58) para que no olvide definir todos los parámetros que requiere cada opción. Esta selección se muestra en la figura (60).

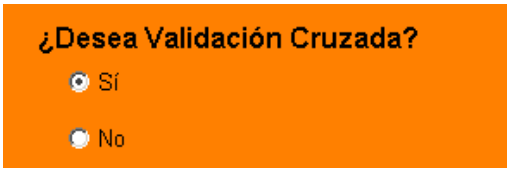


Figura 60: Selección de validación cruzada o no

2. **Zona de selección del tipo de Kernel**. Los Estimadores Kernel son los parámetros más importantes que poseen los modelos de regresión local polinómica implementados en el programa. Su elección será determinante en los resultados que obtengamos. Aparecen reflejados en la figura (61).

Figura 61: Selección del estimador Kernel

3. **Zona de parámetros numéricos que son necesarios para ejecutar el modelo**. Estos parámetros serán o no necesarios dependiendo de si el usuario ha elegido Validación Cruzada o no. Se indican en la figura (62).

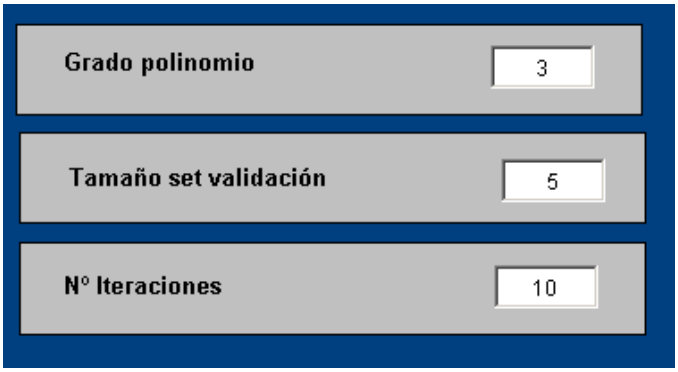


Figura 62: Otros parámetros necesarios.

4. **Zona de selección del Ancho de banda (Band width)**.El segundo parámetro más importante en los modelos no paramétricos implementado en el programa es el ancho de banda. Existen tres formas de elegir el ancho de banda:

   a) **Iguales para todas las variables**. Éste a su vez se divide en:

      1) Valor único: Es el valor que usarán todas las variables y que será el mismo durante la ejecución del modelo. Figura (63).
      2) Intervalo: a través de esta opción, el usuario podrá definir un límite inferior, otro superior y un incremento para el valor del ancho de banda. Esta opción permite muchas más ejecuciones y mucha más afinidad a la hora de examinar los resultados. Figura (64).

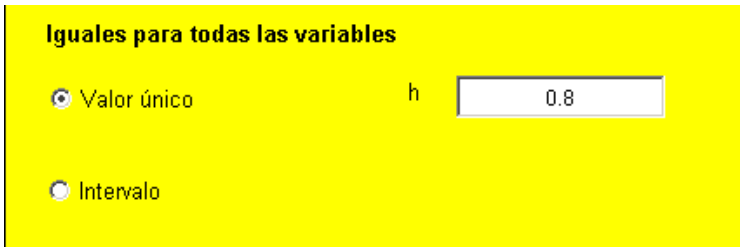


Figura 63: Ancho de banda igual para todas las variables

El usuario debe tener en cuenta, que si elije por ejemplo un límite superior de 0.8, un inferior de 0.7 y un incremento de 0.03, el programa ajustará automáticamente un límite superior de 0.79, para que el programa funcione correctamente. Además, se recomienda no superar las centésimas en el incremento. Un buen valor sería de 0.01 ó 0.02.

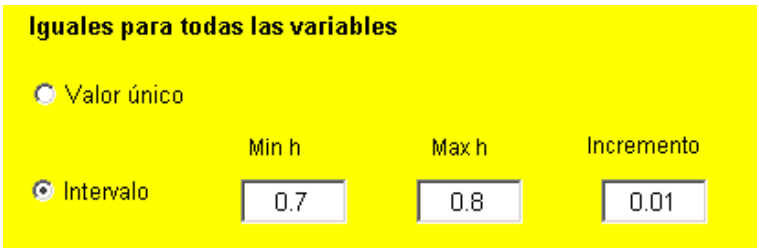


Figura 64: Ancho de banda de intervalo.

b) **Diferentes para todas las variables**. A través de esta opción, el usuario podrá elegir un valor de ancho de banda diferente para cada variable Input. Figura (65).

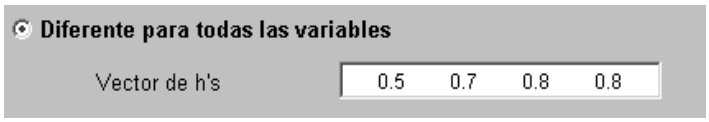


Figura 65: Ancho de banda diferente para todas las variables.

Nota importante: es necesario que la longitud del vector de h's contenido en la figura (65) sea la misma que el número de variables de entrada como indica la figura (66).

5. **Zona que indica el lugar donde están contenidos los datos de entrenamiento y validación**. Para ello, se deben haber cargado antes (como muestra la figura (23)) para indicar dónde están almacenados.

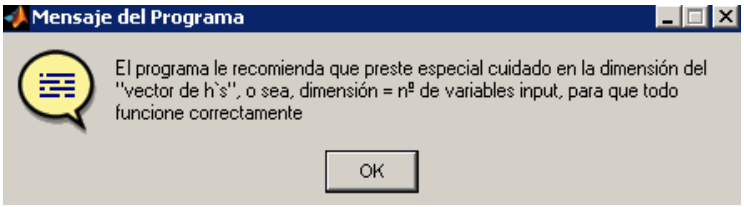


Figura 66: Aviso del programa.

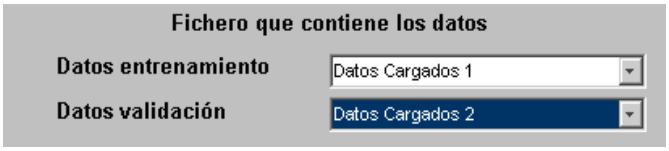


Figura 67: Ubicación de los datos a tratar por el programa.

4.9.4. Ejecución y análisis (Pantalla primera)

En esta pantalla se podrá en práctica la ejecución de los modelos de Regresión Local Polinómica a partir de los parámetros que se hayan indicado en las pantallas previas. Es necesario destacar que la ejecución y resultados del modelo es un proceso iterativo, en el que el usuario prueba con unos determinados parámetros (figura (59)) y después comprueba resultados iterativamente hasta que afina el análisis. Es necesario destacar que este proceso permitirá seleccionar el mejor (permitirá encontrar los parámetros adecuados) que podrá ser probado, luego, con un conjunto de nuevos datos (análisis que se realizará en pantallas posteriores)

Esta pantalla muestra tres posibles versiones, dependiendo si se elije:

1. Validación cruzada activa y ancho de banda único o vector de anchos de banda diferentes. Se muestra en la figura (68).

2. Validación cruzada activa y ancho de banda intervalo. Se muestra en la figura (69).

3. Validación cruzada desactivada y ancho de banda único o vector de anchos de banda diferentes. Se muestra en la figura (70).

4. Validación cruzada desactivada y ancho de banda intervalo. Se muestra en la figura (71).

En las dos primeras configuraciones, usando validación cruzada, figuras (68) y (69)), la interfaz muestra gráficos correspondientes al valor del MARE en porcentaje con respecto al número de iteración realizadas. El promedio es obtenido como la media de todos los errores

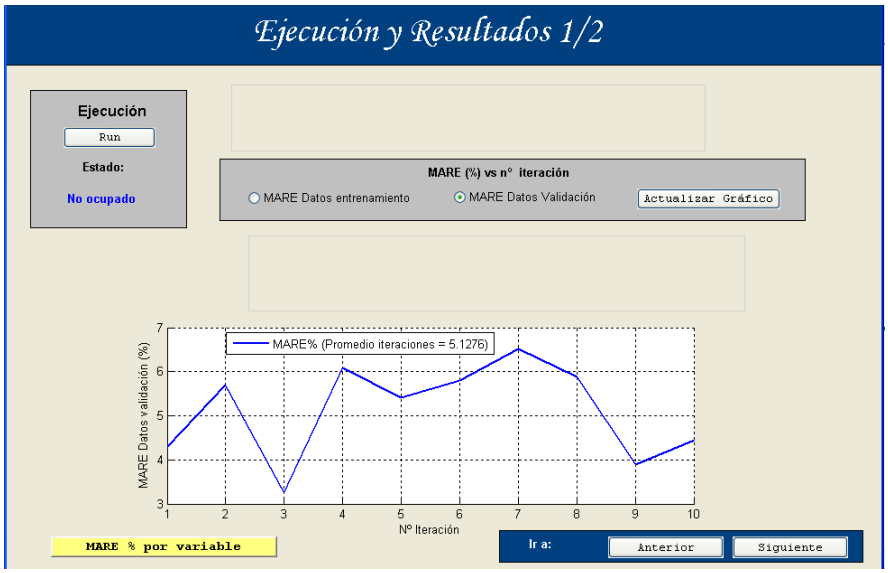


Figura 68: Primera configuración posible para la pantalla

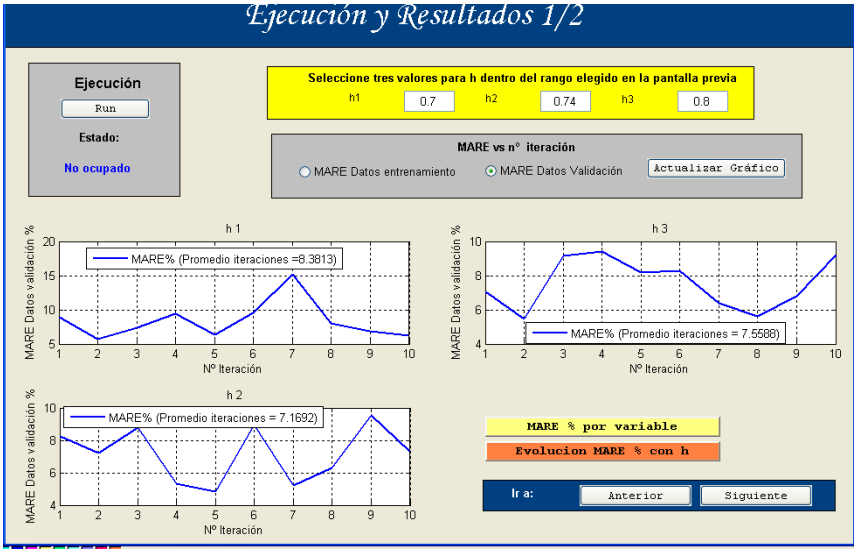


Figura 69: Segunda configuración posible para la pantalla

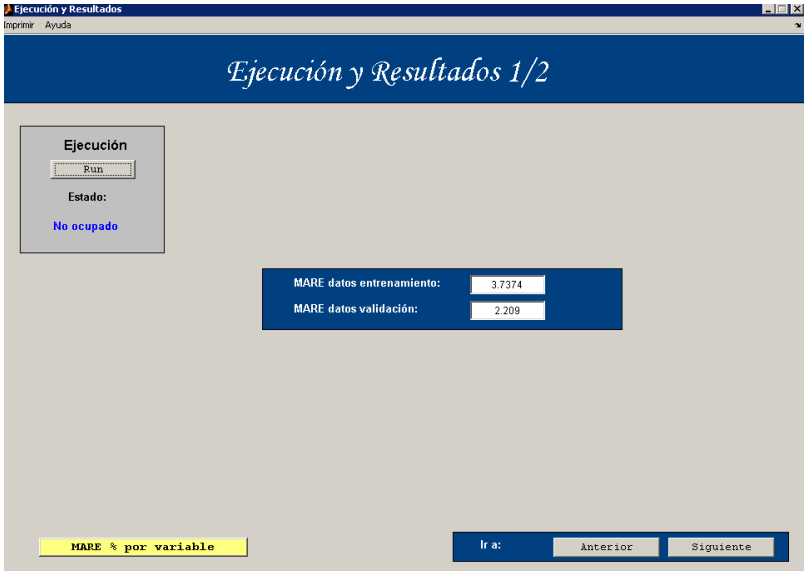


Figura 70: Tercera configuración posible para la pantalla.

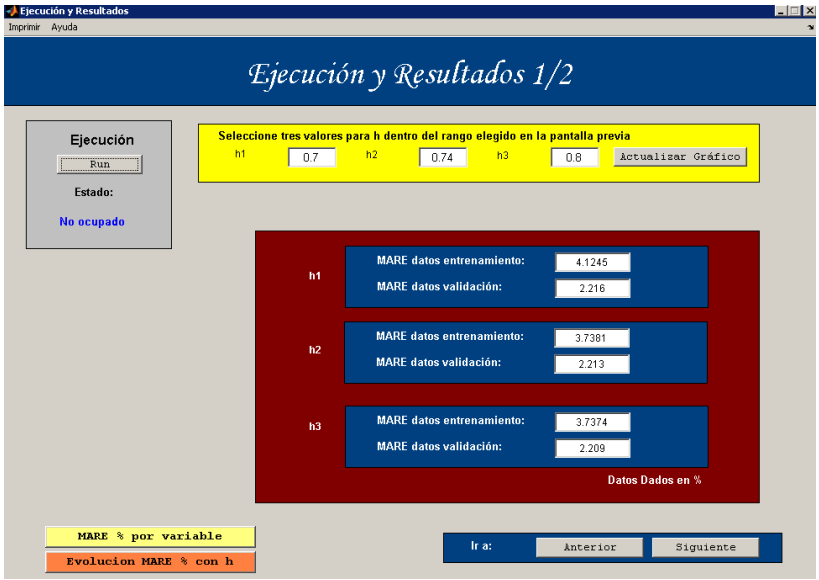


Figura 71: Cuarta configuración posible para la pantalla.

cometidos en las observaciones utilizadas y, para todas las variables salida que se tengan. Además se muestra un recuadro con el valor del MARE obtenido como promedio de todas las iteraciones realizadas.

En las dos siguientes configuraciones, sin usar validación cruzada, figuras (70) y (71)), la interfaz muestra los valores numéricos del MARE, en porcentaje, obtenido en la única iteración realizada. En este caso el promedio usa todas las observaciones y todas las variables de salida

Esta pantalla destaca por su sencillez, debido a que el usuario lo único que deberá realizar es pulsar el botón de "Ejecución que se encuentra.[a] la izquierda de la pantalla y esperar a que el programa le avise que ha terminado de ejecutarse.

Además de estos resultados, la Interfaz, muestra los resultados del MARE para cada variable salida. Para ello, lo único que deberá realizar el usuario es pulsar el botón denominado "**MARE % por variable**"(figura (72)) y acto seguido en la pantalla que aparece pulsar el botón "Ver gráficos"(figura (73)).

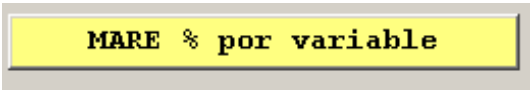


Figura 72: Botón de la pantalla de Ejecución y Análisis.

Figura 73: MARE % en relación a una variable de salida

Si el usuario ha seleccionado un intervalo para el ancho de banda (figura (59)), la Interfaz

mostrará un gráfico de los valores del MARE con respecto a cada valor de ancho de banda. En este caso, el MARE es el promedio obtenido para todas las variables salida y todas las iteraciones realizadas. Este gráfico permite al usuario, conocer qué valor de ancho de banda, produce el mínimo valor de MARE, y por lo tanto, un mejor ajuste del modelo a los datos de entrenamiento y validación. Para ello, el usuario deberá pulsar el botón denominado "**Evolución MARE % con h**"(figura (74)) y pulsar seguidamente "Ver gráficos"(figura (75)).

Figura 74: Botón de la pantalla de Ejecución y Análisis

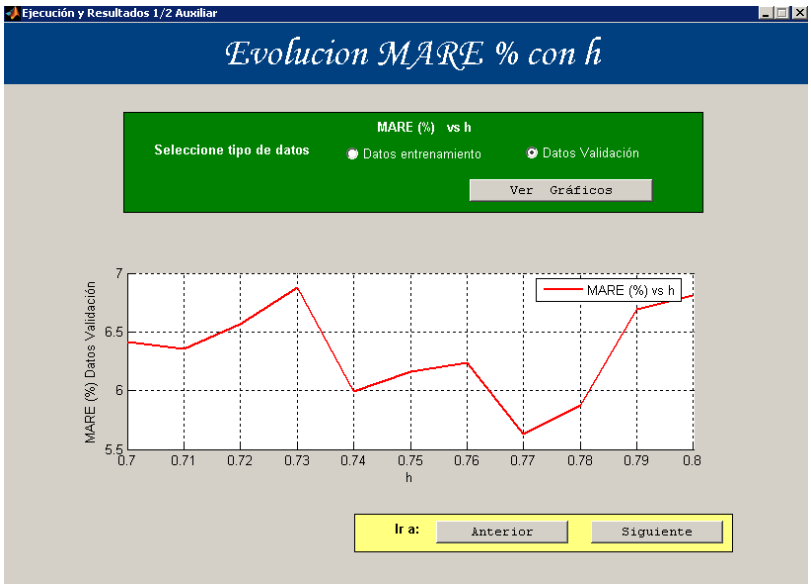


Figura 75: Evolución del MARE % con el ancho de banda.

4.9.5. Ejecución y análisis (Pantalla segunda)

Debido a la cantidad de de valores numéricos y gráficos que se emiten en la ejecución del modelo, se desarrolló esta pantalla, que no es más que la continuación de la primera pantalla de Ejecución y Análisis. El funcionamiento de esta pantalla es totalmente idéntico al de la figura (47). La única diferencia es que en lugar de presentar los resultados del modelo de Regresión Lineal Múltiple, presenta los resultados del modelo de Regresión Local Polinómica. El aspecto general de la pantalla es el que se muestra en la figura (76).

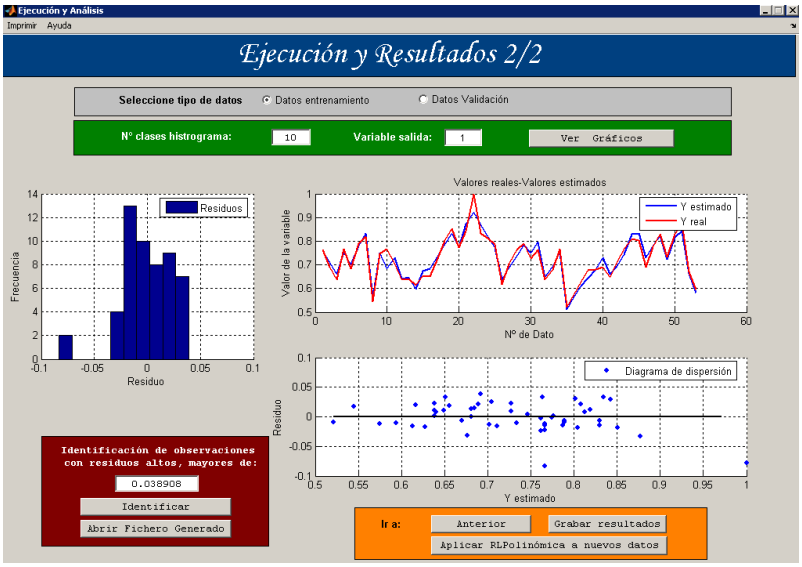


Figura 76: Segunda pantalla de Ejecucción y Análisis.

Un aspecto importante en esta segunda pantalla de Ejecución y Análisis es que permite acceder a la pantalla de Guardar Datos. Para acceder a la pantalla de guardar datos, debe pulsar el botón que aparece en la figura (77).

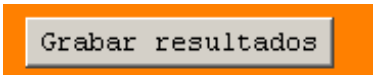


Figura 77: Botón para acceder a la pantalla de Guardar Resultados.

4.9.6. Guardar Resultados

Después de realizar el análisis para seleccionar los parámetros del modelo, el programa permite al usuario guardar los resultados obtenidos El aspecto general de la pantalla se muestra en la figura (78).

El esquema que presenta la pantalla es muy sencillo. En ella, se pretende que se guarden tres tipos de archivos diferenciados:

1. **Archivo 1º**: se indican los parámetros que se han utilizado para ejecutar el programa. El fichero generado es el siguiente:

   a) Parametros_Seleccionados(Archivo 1º).txt

2. **Archivo 2º**: se indica el MARE % por iteración tanto en los datos de entrenamiento

Figura 78: Pantalla Grabar Resultados

como en los datos de validación. Los datos se muestran para los anchos de banda utilizados. Cuando se guardan, se generan dos ficheros, que son:

a) MARE_por_iteracion_train(Archivo 2^o).txt

b) MARE_por_iteracion_validacion(Archivo 2^o).txt

3. **Archivo 3^o**: se guardan los valores de los datos de entrada al modelo, datos de salida con transformación (si se ha usado) y datos de salida estimados con transformación. Se generan tanto para los datos de entrenamiento como para los datos de validación. Otro aspecto importante es que sólo se guardan los valores de la última iteración y para el ancho de banda con valor más elevado. Los ficheros que se guardan son:

   a) Datos_entrada_salida_entrena(Archivo 3^o).txt

   b) Datos_entrada_salida_validación(Archivo 3^o).txt

En este caso, lo único que deberá realizar el usuario es pulsar el botón "Guardar Datosτ esperar que aparezca el mensaje emitido por el programa mostrado en la figura (79).

4.9.7. Pasos a seguir para encontrar los parámetros óptimos del modelo de Regresión Local Polinómica

Introducción al proceso de búsqueda Como se ha comentado anteriormente, para conseguir que el programa pueda estimar nuevos datos a partir de otros de entrada es necesario

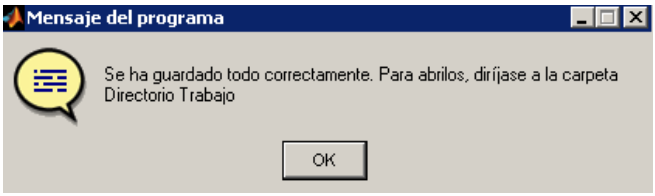


Figura 79: Mensaje del programa

que el modelo esté a punto. Para ello, el usuario deberá entrenarlo con unos valores (datos de entrenamiento y validación) y unos parámetros propios del modelo. Cambiando los parámetros y ejecutando el modelo tantas veces como sea necesario, se conseguirá poner a punto el programa para garantizar al usuario que los datos que seguidamente se estimarán tendrán el mínimo error posible. Este proceso iterativo se muestra en la figura (80).

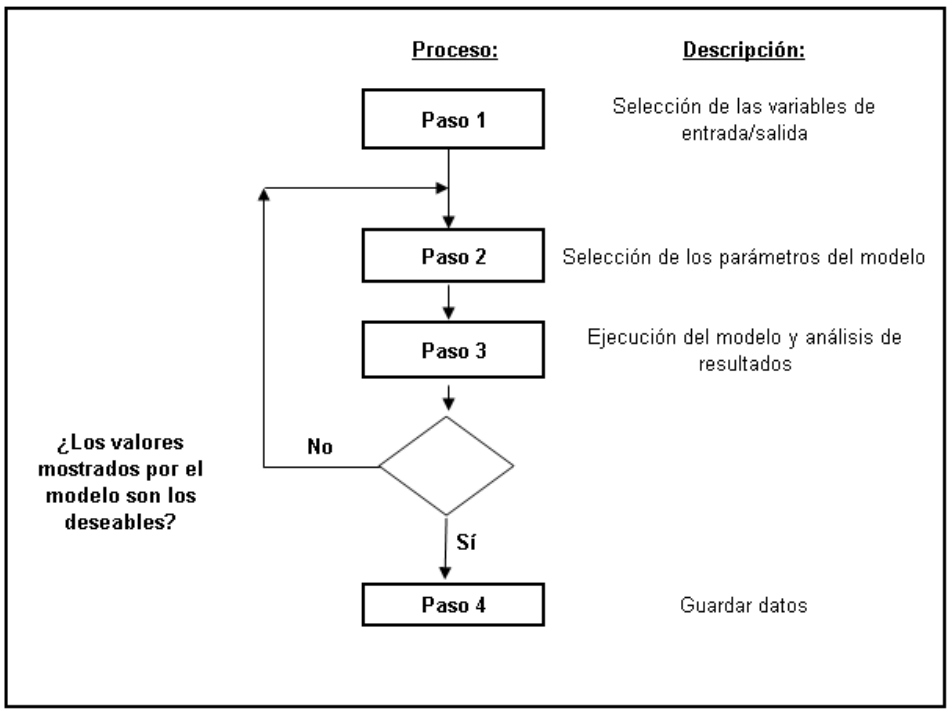


Figura 80: Proceso iterativo de puesta a punto del modelo.

Paso 1 Para comenzar, el usuario debe situarse en la pantalla de selección de variables de entrada/salida (figura (55)). En esta pantalla, el usuario tendrá la opción de elegir las variables y una posible transformación de los datos (figura (56)). Para seleccionar las variables de entrada/salida, el usuario puede hacer uso del análisis de previo de datos (figura (30)). De

esta forma le será más sencillo conocer cuántas columnas (variables totales posibles) posee el archivo de datos. Una vez que el usuario haya escrito los valores de las columnas que quiere tomar como variables de entrada/salida, debe pulsar el botón "**Introducir Información Al Programa**"

Por otra parte, se ha dejado una posible configuración por defecto, siendo muy útil si el usuario no tiene claro qué variables escoger.

Para continuar al segundo paso, deberá pulsar el botón "**Parámetros modelo**"

Paso 2 Una vez que el usuario ha pulsado el botón "Parámetros modelo"se encontrará en la pantalla de Selección Parámetros (figura (59)). En esta pantalla lo que el usuario deberá realizar es escoger unos parámetros adecuados para la ejecución del modelo.

En un primer uso de la selección de parámetros, se recomienda que el usuario deje los valores por defecto. En la próxima ocasión que el usuario se encuentre en esta pantalla, se recomienda que cambie un sólo parámetro y evalúe los cambios en las siguientes pantallas, debido a que si se realiza de esta forma, es más fácil detectar los cambios que provoca el cambio en un determinado parámetro.

Otra recomendación importante es que el programa puede que no funcione bien si se le pide que trabaje en un rango de ancho de banda muy bajo, por ejemplo, cercano a 0.4 o menor, debido a que puede que no caiga más de un dato dentro de un intervalo tan pequeño. Si el usuario elige un valor pequeño y el programa detecta que no es adecuado para las condiciones de trabajo, mostrará un mensaje de aviso (figura (81)).

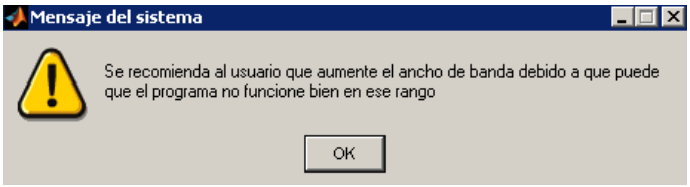


Figura 81: Aviso del programa.

Para continuar a la pantalla de Ejecución y Análisis, debe pulsar el botón "**Siguiente**".

Paso 3 Una vez que el usuario ha seleccionado los parámetros deseados para el modelo, debe ejecutarlo para obtener y analizar los resultados que se mostrarán en tablas y gráficos. Para ello, una vez que el usuario se encuentre en la pantalla de "Ejecución y Análisis"(cualquiera de sus múltiples versiones, por ejemplo la mostrada en la figura (68)) debe pulsar el botón de 'Run' mostrado en la figura (82).

En el momento que el usuario pulse el botón '**Run**' aparecerá una pantalla que muestra un pequeño resumen de los parámetros que se han introducido al modelo, como el que muestra

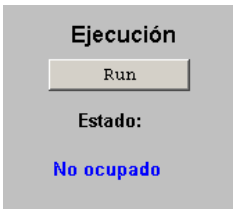


Figura 82: Botón que permite ejecutar el modelo.

la figura (83) y el estado del programa pasará a estar en la situación de 'Ocupado'.

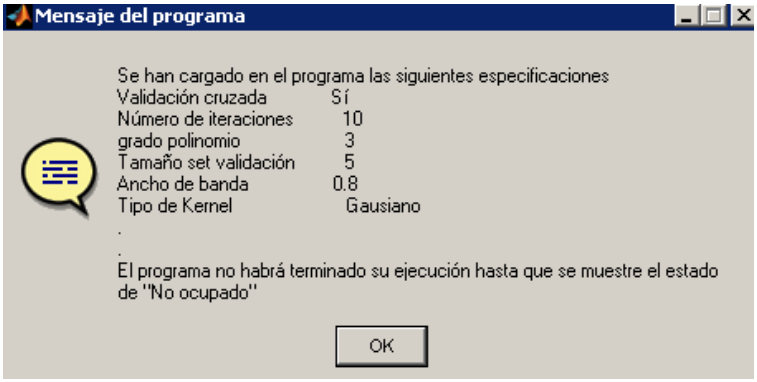


Figura 83: Resumen de los parámetros elegidos.

Aunque la pantalla haya mostrado el estado 'Ocupado', el programa no comenzará a ejecutarse hasta que el usuario no pulse el botón '**Ok**' que muestra la figura (83). El mensaje de aviso (mostrado en la figura (83)) no desaparecerá hasta que el modelo se haya ejecutado con éxito, por lo tanto, se recomienda que el botón 'Ok' no se pulse más de una vez para no saturar el programa.

Una vez que el programa se haya ejecutado con éxito, se retirará el mensaje de aviso (figura (83)) y se mostrará un mensaje del programa como el que muestra la figura (84).

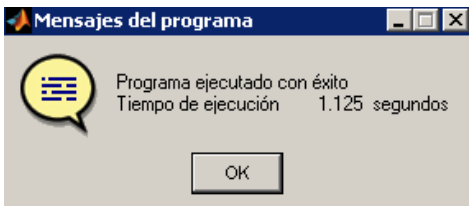


Figura 84: Mensaje que indica que el modelo se ha ejecutado con éxito.

Inmediatamente que el programa se haya ejecutado se mostrarán datos del MARE% (Error absoluto relativo medio) para los datos de entrenamiento o validación. Estos datos se

pueden mostrar en forma de gráfico (si el usuario eligió Validación cruzada) o un dato único (sin Validación cruzada).

En este punto, el usuario puede aplicar su criterio en base al valor del MARE % para decidir si debe cambiar los parámetros del modelo (por ejemplo, si le parece que el valor del MARE % es muy elevado). Si los valores de MARE % no le ayudan a decidirse, puede continuar a la siguiente pantalla (pulsando '**siguiente**') donde se muestran muchos más gráficos de análisis (figura (76)). Se entiende que con todos estos datos y gráficos el usuario tendrá el criterio suficiente para decidir si volver atrás para reajustar los parámetros del modelo (figura (59)) o considerar que el modelo está suficientemente afinado para poder estimar nuevos datos. De todas formas, se recomienda que la primera vez que se ejecute el modelo se reajusten los párametros para ver si el valor de MARE % mejora.

Paso 4 Una vez que el usuario ha conseguido afinar el modelo en base a cambiar los parámetros que lo gobiernan, podrá guardar los datos de este análisis. Para guardarlos, debe pulsar el botón '**Grabar resultados**' que se encuentra en la pantalla mostrada en la figura (76) y pulsar el botón '**Guardar Datos**' que se encuentra en la pantalla de guardar (figura (78)) .

4.10. Aplicar Regresión Local Polinómica a nuevos datos

Esta opción permite aplicar RLP a un nuevo conjunto de datos, usando unos parámetros específicos, que pueden ser los seleccionados como mejores en la etapa previa. Para ello, el usuario debe cargar un conjunto nuevo de datos, conteniendo sólo los valores de las variables de entrada.

El formato que presenta la pantalla de estimación de nuevos datos es el que se muestra la figura (85).

Para acceder a esta sección del programa, deberá pulsar el botón '**Aplicar RLPolinómica a nuevos datos**' en la segunda pantalla análisis (76)

Un aspecto importante para acceder a esta pantalla, es que el usuario haya guardado los datos de entrada que desea utilizar para estimar nuevos datos en la ranura 3 de la pantalla de carga de datos (23), de lo contrario el programa mostrará un mensaje de aviso y no le dejará continuar.

4.10.1. Organización de la pantalla

La pantalla consta de cuatro partes:

1. **Selección del fichero de datos y las variables de entrada/salida**: Es la zona donde el usuario podrá seleccionar el archivo que contiene los datos y las variables

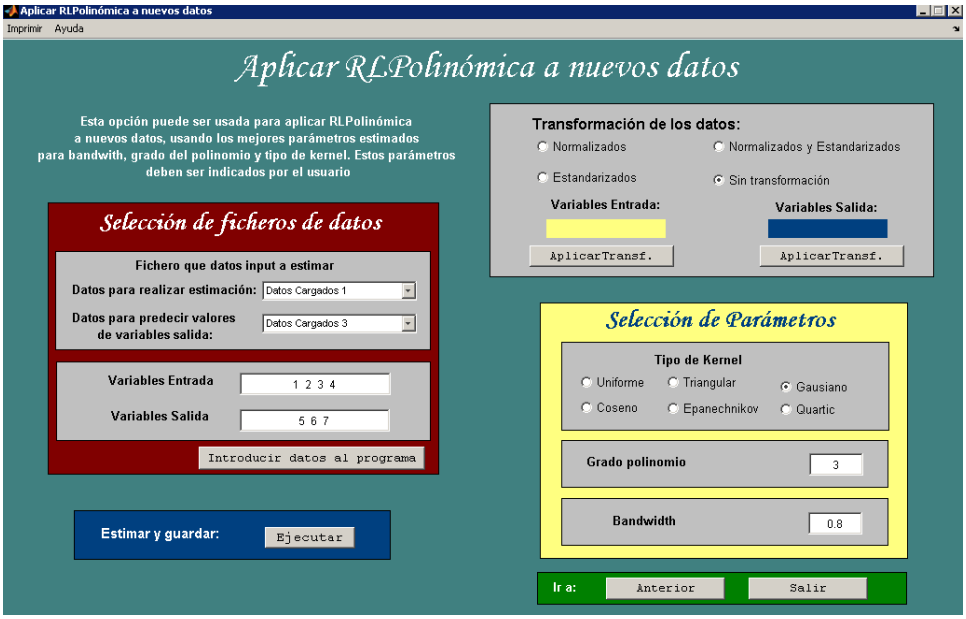


Figura 85: Aplicación del modelo a nuevos datos

de entrada/salida. Se recomienda al usuario que si ha puesto a punto el modelo, por ejemplo, a partir de 4 variables de entrada, que el archivo de datos de entrada a estimar tenga también 4 variables. De todas formas, se da la opción de seleccionar las que el usuario desee. Esta sección se muestra en la figura (86).

2. **Aplicar una posible transformación a los datos**. De forma separada, se pueden aplicar dos transformaciones diferentes, dependiendo si son variables de entrada o variables de salida. Las posibles transformaciones se muestran en la figura (87).

3. **Selección de parámetros del modelo**. Por defecto, se mostrarán los últimos parámetros que el usuario utilizó en la pantalla de selección de parámetros (figura (59)). Esta zona de la pantalla se muestra en la figura (88).

4. **Zona de ejecución del modelo**, mostrado en la figura (89).

4.10.2. Pasos para Aplicar RLP a un nuevo conjunto de datos

Paso 1 Lo primero que deberá realizar el usuario será seleccionar será la ubicación de los datos que han servido para entrenar el modelo y la ubicación de los datos de entrada de los que se quieren proporcionar nuevos datos (figura (86)). Una vez que el usuario haya seleccionado los archivos y las variables, deberá pulsar el botón '**Introducir datos al programa**'. Seguidamente aparecerá el mensaje del programa mostrado en la figura (90).

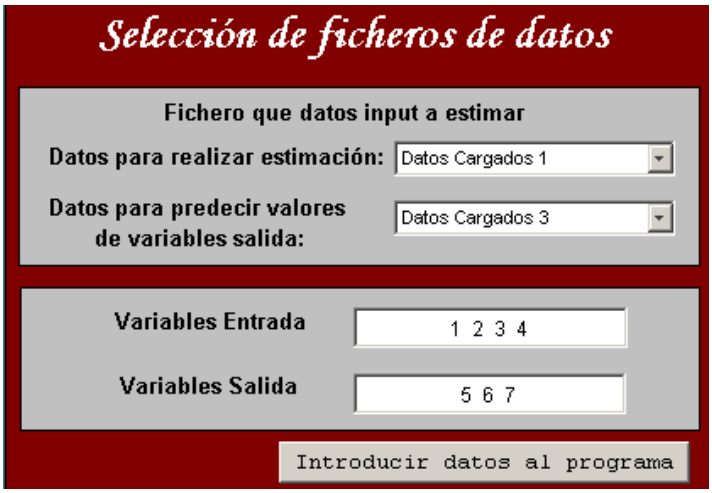


Figura 86: Selección de ficheros de datos

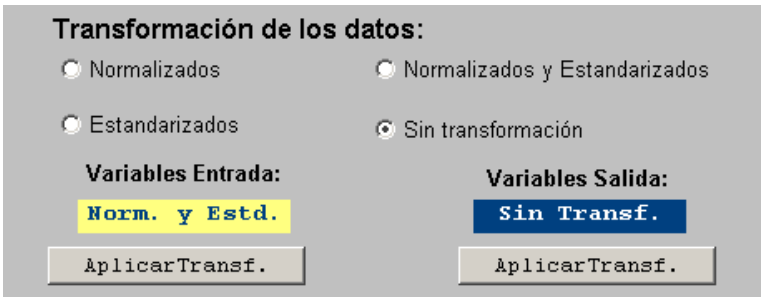


Figura 87: Posibles transformación a las variables.

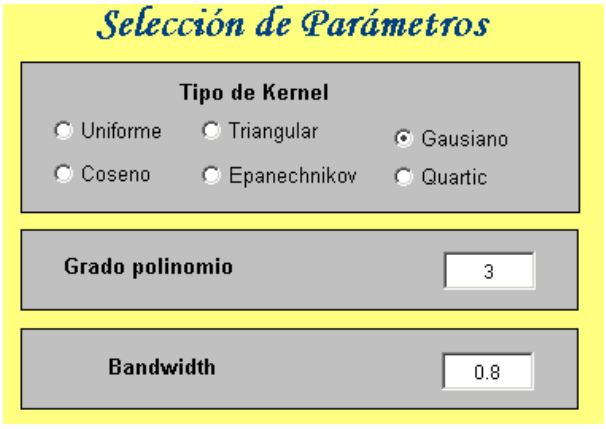


Figura 88: Selección de parámetros del modelo.

Figura 89: Ejecución del modelo.

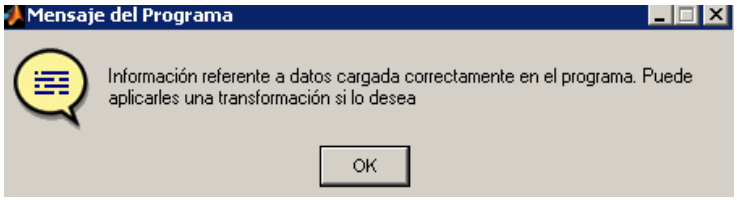


Figura 90: Mensaje del programa

Paso 2 Para continuar con el proceso el usuario deberá seleccionar una posible transformación a las variables de entrada y salida por separado (figura (87)). Si el usuario no desea aplicar ninguna transformación, deberá seleccionar la opción 'Sin Transformación'. Seguidamente debe pulsar los botones de 'Aplicar Transf.' En este mismo instante, el programa emitirá un mensaje como el mostrado en la figura (91).

Paso 3 Una vez que se han realizado los pasos 1 y 2 correctamente (se han mostrado los correspondientes mensajes del programa verificándolo), el usuario deberá seleccionar los parámetros oportunos (por defecto se muestran los últimos seleccionados) y pulsar el botón '**Ejecutar**' mostrado en la figura (89).

Cuando el programa termine de ejecutar el modelo, se mostrará el mensaje de la figura (92).

Los datos estimados se guardarán en el archivo '**Datos_nuevos_estimados(Archivo 4º).txt**', ubicado en la carpeta '**DirectorioTrabajo**'.En este punto, el programa habrá terminado sus funciones y el usuario podrá pulsar el botón '**Salir**'.

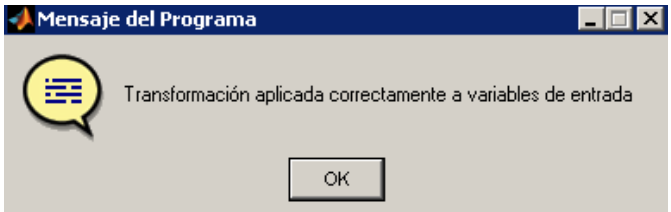


Figura 91: Mensaje del programa

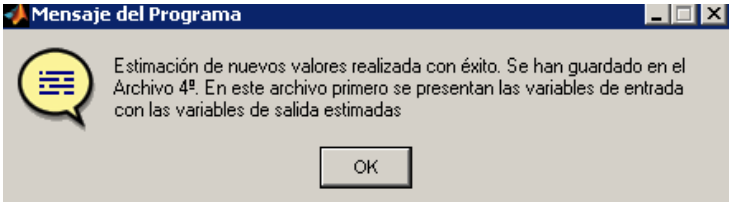


Figura 92: Mensaje del programa

5. Aplicación de la Interfaz MathNonParametrics a Datos Reales

5.1. Introducción

Una vez que se ha descrito la teoría sobre Regresión Local Polinómica, así como, su implementación en la interfaz gráfica 'MathNonParametrics', en este capítulo se ilustrará el uso de esta metodología mediante la interfaz desarrollada, aplicándola a 2 ejemplos reales. El objetivo es mostrar el uso de la herramienta (más que el obtener el mejor modelo de regresión local posible).

El primer ejemplo consiste en modelizar la curva de potencia de un aerogenerador eólico. El segundo ejemplo corresponde a la modelización de fuerzas de mecanizado, en un proceso de taladrado.

5.2. Curva de potencia de un aerogenerador

5.2.1. Producción de potencia

La curva de potencia de un aerogenerador, se define de manera general como la relación entre la potencia eléctrica producida por un aerogenerador para diferentes velocidades del viento. En la figura (93) se muestra un ejemplo de una curva de potencia. Sin embargo, hay que tener en cuenta que la potencia producida por el aerogenerador depende también de otras variables, como por ejemplo, la dirección del viento, la temperatura, etc. Por tanto, se podría obtener un modelo multivariante para dicha potencia.

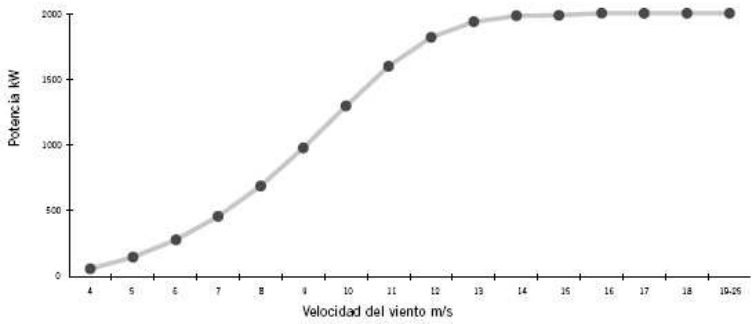


Figura 93: Forma común de la curva de potencia de un aerogenerador

Para obtener de forma experimental una curva Potencia-velocidad de viento, es necesario realizar medidas en campo. Para ello un anemómetro es situado sobre un mástil relativamente cerca del aerogenerador, de manera que se puedan tomar medidas fiables (no sobre el

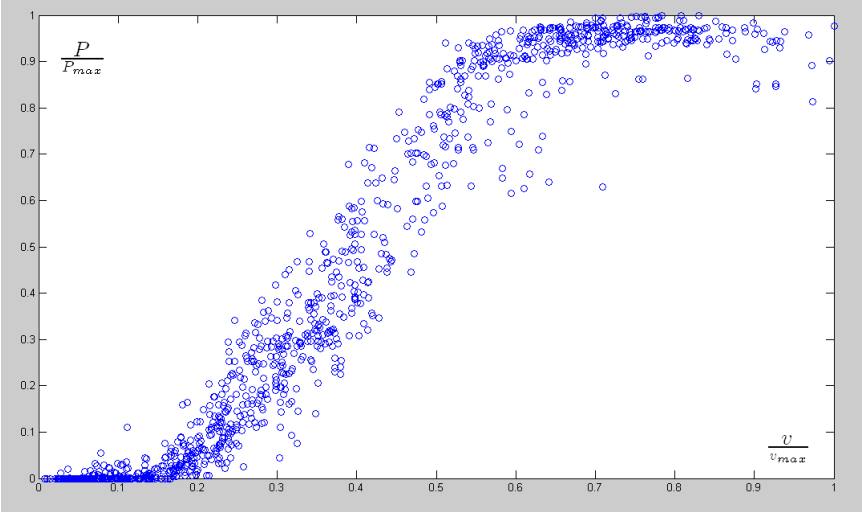


Figura 94: Curva de potencia-velocidad de viento experimental.

mismo aerogenerador ni demasiado cerca de él, pues el rotor del aerogenerador puede crear turbulencia, y hacer que la medida de la velocidad del viento sea poco fiable).

En la figura (94), se representa el conjunto de medidas tomadas para un aerogenerador, el cual será usado en la modelización con Regresión Local Polinómica. Como se puede ver, la curva potencia-velocidad no es una línea perfecta (como en la figura (93). Por el contrario, existe dispersión en los valores, es decir, para el mismo valor de viento se obtienen distintos valores de potencia.

En nuestro caso, disponemos de valores de velocidad de viento y dirección de viento. Por ello, se intentará modelizar la curva de potencia en función de estas dos variables explicativas.

5.2.2. Estructura del fichero que contiene los datos

El fichero de datos que se utilizará para realizar el análisis, 'Corral.txt', consta de tres variables que se ordenan como sigue:

- Primera columna se dan datos de potencia en W
- Segunda columna se dan datos de velocidad del viento en m/s
- Tercera columna se dan datos de dirección de incidencia del viento, en radianes.

Se dispone de un total de 1083 datos.

El objetivo es estimar un modelo que permite predecir la potencia en función de la velocidad y dirección del viento. A continuación se muestran los resultados obtenidos al aplicar Regresión Local Polinómica.

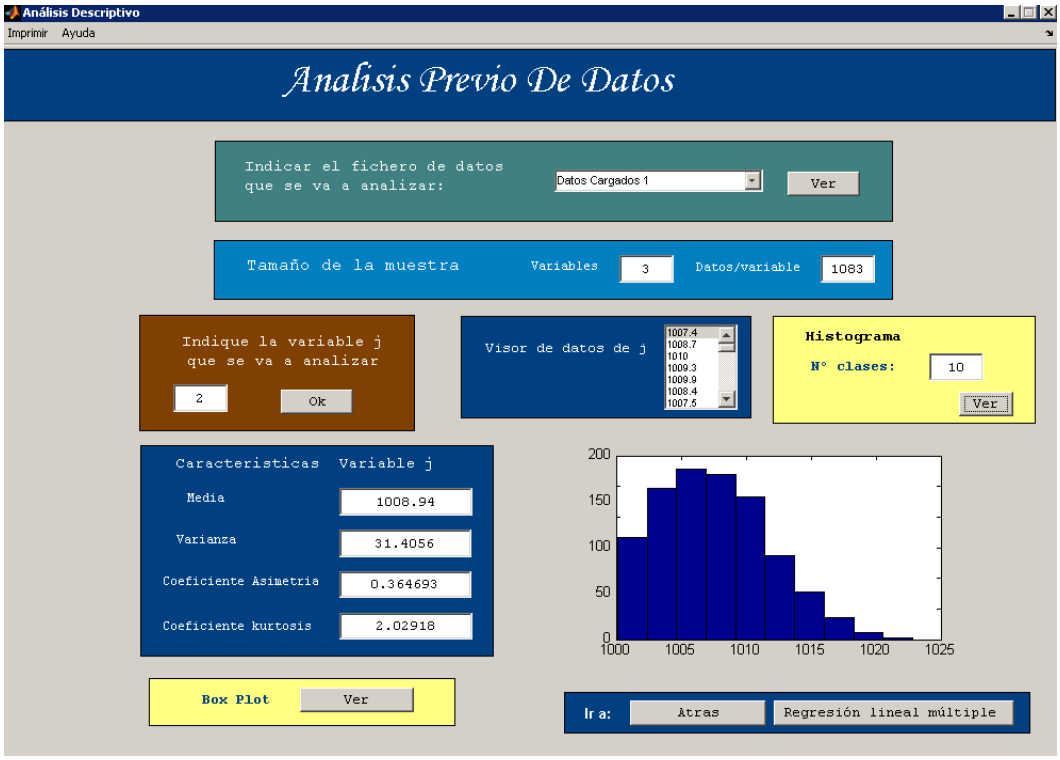


Figura 95: Pantalla Análisis previo de datos

5.2.3. Análisis Previo de Datos

El primer paso, es realizar un análisis descriptivo univariante de las variables. Para ello se usa la opción 'Análisis Previo de Datos' que proporciona la interfaz MathNonParametrics. Mediante este análisis podemos ver, por ejemplo, si las variables tienen distribuciones próximas a la normal, si son simétricas o no, si hay muchos valores extremos, si tienen mucha o poca dispersión.

Por ejemplo, a continuación se muestran los resultados de este análisis para la segunda variable del archivo, que corresponde a la velocidad el viento en m/s. Estos resultados se muestran en las figuras (95), histograma (96) y boxplot (97).

Se puede concluir que la variable velocidad de viento sigue una distribución ligeramente asimétrica hacia la derecha, y no parecen existir valores extremos o atípicos.

Una vez que realizado el análisis univariante, el siguiente paso es aplicar un modelo de Regresión Lineal Múltiple, el cual se muestra a continuación.

5.2.4. Regresión Lineal Múltiple

En la pantalla que brinda la interfaz para ejecutar este modelo, se van a introducir como variables de entrada las número 2 y 3 (velocidad viento y dirección), y como variables de

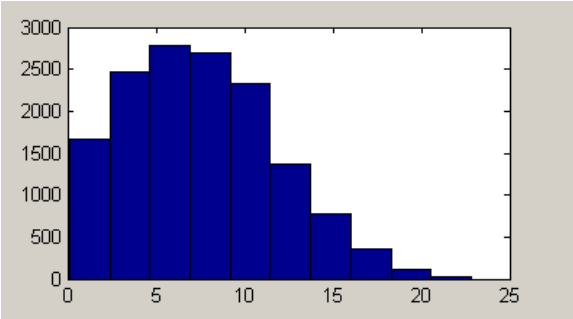


Figura 96: Histograma de la velocidad del viento en m/s

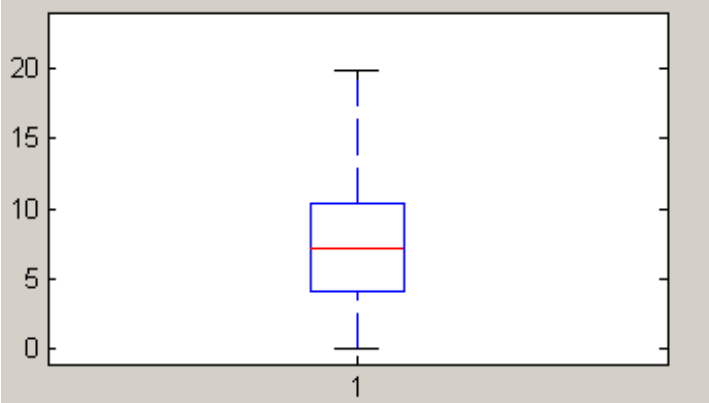


Figura 97: Diagrama de caja de la velocidad del viento en m/s

salida la número 1 (potencia generada).

Una vez que se ha ejecutado el modelo, se observan los siguientes resultados para los valores de potencia:

- Valores reales-Valores estimados, figura (98).
- Diagrama de dispersión Yestimado-Residuo, figura (99).
- Histograma de residuos, figura (100).

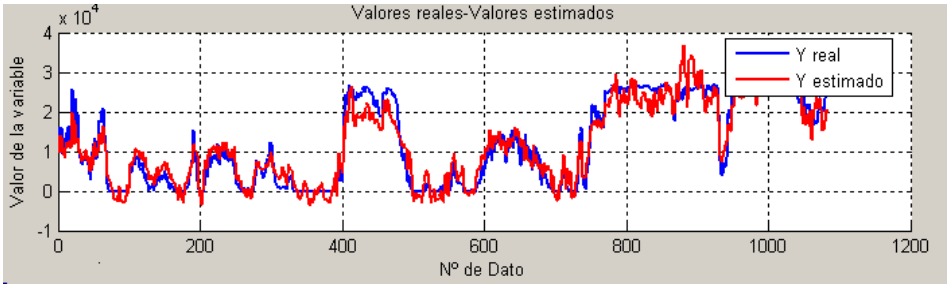


Figura 98: Gráfico Valores reales-Valores estimados

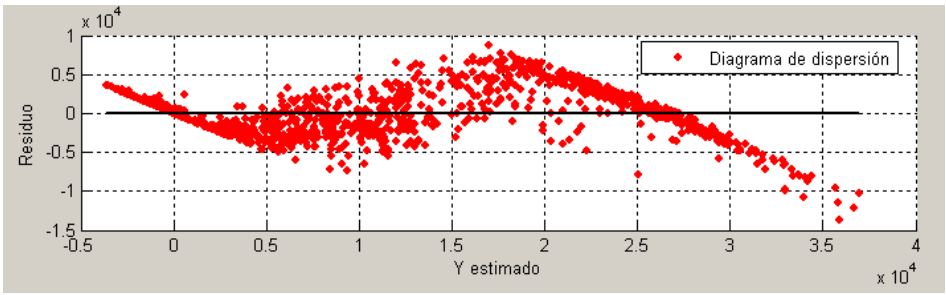


Figura 99: Gráfico Yestimado-residuo

A partir de estos resultados, se puede concluir que el modelo lineal no es una buena aproximación. Como se observa en la figura (99), los residuos muestran claramente una relación no lineal que no ha sido tenida en cuenta por el modelo lineal. En este gráfico no se observan residuos con valores muy altos, en valor absoluto, de manera que no se ha realizado una identificación de posibles valores atípicos.

5.2.5. Regresión Local Polinómica

Lo primero que solicitará la interfaz MathNonParametrics para este modelo es que se le introduzcan las variables de entrada y salida, que se eligirán de forma análoga al modelo de

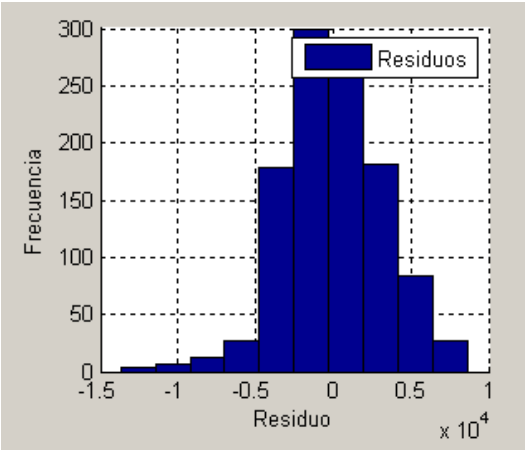


Figura 100: Histograma de residuos

Regresión Lineal Múltiple. Además, cuando se inserte este dato en el programa, se eligirá una posible transformación de los datos. En nuestro caso, se elegirá la opción de 'Normalizados', debido a que las variables poseen diferentes órdenes de magnitud entre sí.

El siguiente paso será el de elegir el tamaño del set de validación que va a servir para obtener los gráficos que se mostrarán más adelante. Como el archivo de datos posee 1083 datos, se tomarán para la validación alrededor de un 15 % de ellos, es decir, unos 163 datos. Por lo tanto, quedarían 920 datos para realizar el entrenamiento.

Para seleccionar el mejor modelo se deben encontrar cuáles son los mejores valores de los siguientes parámetros:

- función Kernel
- grado del polinomio
- ancho de banda.

En primer lugar, se han realizado distintas pruebas variando el tipo de Kernel. Los resultados muestran que no existe mucha diferencia entre usar un tipo de Kernel u otro, de manera que en el kernel que va a ser usado en el modelo es el **Kernel Gaussiano**. A continuación se ha buscado el grado del polinomio que proporcione los mejores resultados. En este caso, la mejor opción es usar **grado de polinomio igual a 3**. Para la selección de estos valores se ha usado la opción de Validación Cruzada.

Para seleccionar el mejor valor del ancho de banda, se sigue el mismo procedimiento. Usando Validación Cruzada se buscará el ancho de banda que proporciona el menor error de predicción, en los datos de validación. Para facilitar la búsqueda se elige la opción de intervalo para el ancho de banda. dado que las variables de entrada están normalizadas, se usará el

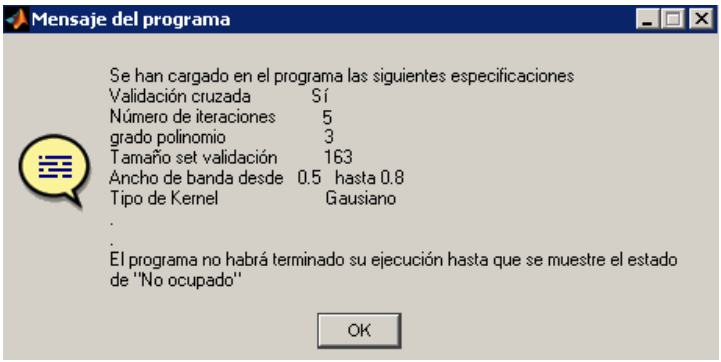


Figura 101: Parámetros en la ejecución del modelo.

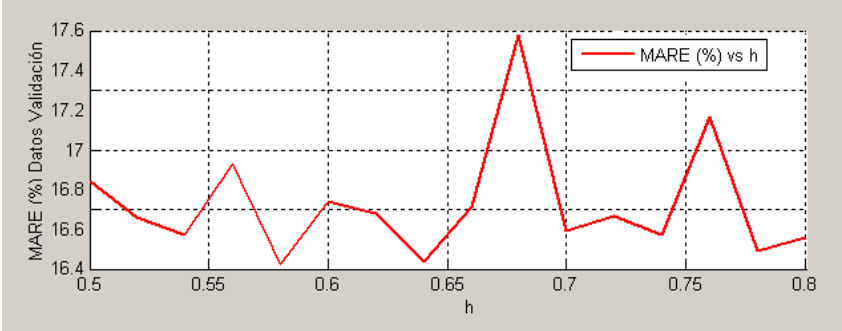


Figura 102: Evolución del MARE % con el ancho de banda

mismo valor de ancho de banda para todas ellas. El número de iteraciones realizadas, por cada ancho de banda, es igual a 5.

Como resumen, los valores de los parámetros utlizados (tipo de kernel, grado del polinomio, número de iteraciones, etc) aparecen indicados en la figura (101).

A continuación se muestran los resultados obtenidos al ejecutar regresión local para anchos de banda comprendidos en el intervalo [0.5,0.8]. En la figura (102) (opción de 'Evolución de MARE % con h'), se observa que el mejor valor del ancho de banda es cercano a 0.65 Para ese valor se obtiene el menor error de predicción MARE, en los datos de validación, igual a 16 % .

Además del error de predicción esperado, el programa proporciona otros gráficos para la variable salida(gráficos de datos normalizados entre cero y uno de potencia):

- Valores reales-Valores estimados, figura (103).
- Diagrama de dispersión Yestimado-Residuo, figura (104).
- Histograma de residuos, figura (105).

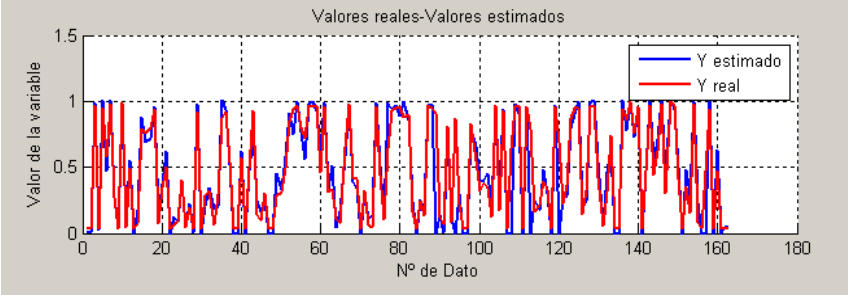


Figura 103: Gráfico Yestimado - Yreal

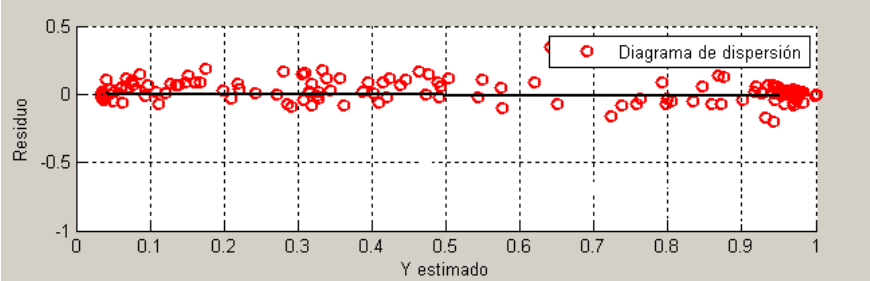


Figura 104: Diagrama de dispersión Residuo - Yestimada

Estos gráficos corresponden a la última iteración ejecutada para el ancho de banda 0.65. Se ha decidido poner los gráficos de la última iteración debido a que al ejecutar varias veces, se obtienen comportamientos similares, y por lo tanto el razonamiento es válido.

A partir de los resultados mostrados en las figuras (103), (104) y (105), se puede concluir que la bondad de ajuste del modelo es aceptable. En en el gráfico de residuos se observa todavía cierto comportamiento no lineal. Sin embargo, los valores reales y estimados son bastante similares. Si se comparan las figuras (98) (Modelo de Regresión Lineal Múltiple) y (103) (RLPolinómica), se observa qué el modelo de RLP proporciona una mejor bondad de ajuste. Con el fin de mejorar el modelo, quizás se puede introducir otra variable explicativa, como por ejemplo,la temperatura o la presión.

5.2.6. Estimación de nuevos datos a partir del modelo de RLPolinómica

En esta sección se estimará el valor de la potencia para un nuevo conjunto de 100 datos con valor comprendido entre cero a uno. El modelo de regresión local que será aplicado tiene los siguientes parámetros:

- Función Kernel: Gaussiano
- Grado del polinomio: 3

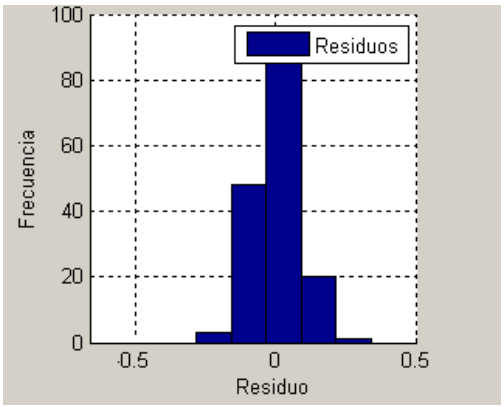


Figura 105: Histograma de residuos

- Ancho de banda para las dos variables de entrada: 0.65

Para mostrar los resultados de este análisis, en la figura (106) se representa la potencia estimada frente a valores de velocidad de viento mediante una línea continua. En este caso se conocen los valores reales de potencia, de manera que estos valores se representan también en la figura. Se observa como el modelo se va ajustando de forma óptima a los datos.

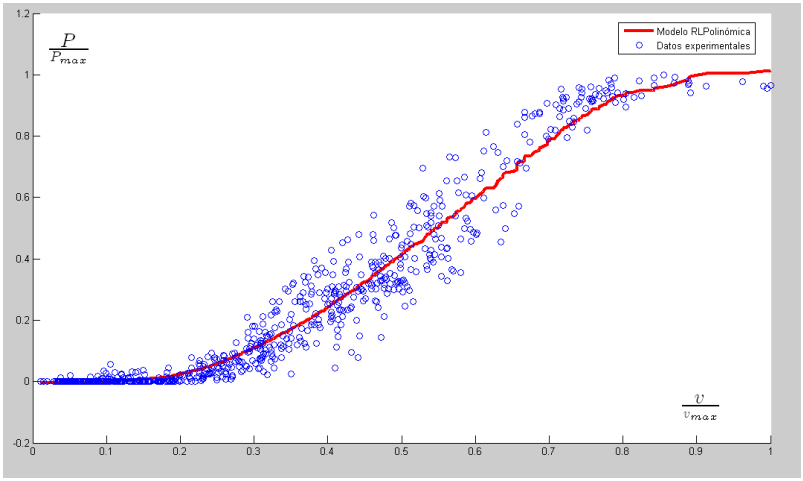


Figura 106: Curva de potencia del aerogenerador obtenida mediante la interfaz MathNon-Parametrics

Figura 107: Máquina de taladrar

5.3. Proceso de fabricación: Taladrado

5.3.1. Introducción

El proceso de taladrado de taladrado es un término que cubre todos los métodos para producir agujeros cilíndricos en una pieza con herramientas de arranque de viruta. Además del taladrado de agujeros cortos y largos, también cubre el trepanado y los mecanizados posteriores tales como escariado, mandrinado, roscado y brochado. El proceso se ilustra en la figura (107).

Con el desarrollo de brocas modernas el proceso de taladrado ha cambiado de manera drástica, porque con las brocas modernas se consigue que un taladro macizo de diámetro grande se pueda realizar en una sola operación, sin necesidad de un agujero previo, ni de agujero guía, y que la calidad del mecanizado y exactitud del agujero evite la operación posterior de escariado.

El objetivo del ejemplo que se presenta aquí, es poder predecir la Fuerza de corte y la Fuerza axial durante el taladrado, para diferentes valores de los siguientes parámetros de mecanizado: ángulo α, ángulo γ, velocidad corte (m/min), avance (mm/rev) profundidad pasada (mm).

5.3.2. Estructura del fichero que contiene los datos

El fichero donde se albergan los datos ('DatosFuerzas11.txt') contiene diecinueve columnas y 150 datos. Las variables que van a ser utilizadas en este ejemplo son las siguientes:

- Décima segunda columna: Ángulo α (radianes)
- Décima cuarta columna: Ángulo γ (radianes)
- Décima quinta columna: Velocidad corte (m/min)
- Décima sexta columna: Avance (mm/rev)
- Décima séptima columna: Profundidad pasada (mm)
- Décima octava columna: Fuerza corte (N)
- Décima novena columna: Fuerza axial (N)

Las demás columnas corresponden a información complementaria.
En el análisis realizado en la interfaz, se eligirán las siguientes variables de entrada/salida:

- Entrada: Ángulo α (radianes), Ángulo γ (radianes), Velocidad corte (m/min), Avance (mm/rev), Profundidad pasada (mm)
- Salida: Fuerza corte (N), Fuerza axial (N).

5.3.3. Análisis Previo de Datos

De forma análoga a como se realizó en el ejemplo de la curva de potencia del aerogenerador, se puede hacer uso de la pantalla de Análisis Previo de Datos que incorpora la interfaz gráfica MathNonParametrics para conocer cómo son los datos y aplicarles un primer análisis estadístico. Este análisis permite conocer el tipo de distribución de los datos, posibles valores atípicos, datos numéricos de coeficientes de asimetría,etc. Para ilustrar como se ha utilizado esta pantalla, se tomará la variable de Fuerza Corte (N) (primera variable de salida), el análisis se muestra en la figura (108).

A la vista de estos resultados,se puede concluir que la variable Fuerza de Corte (N) sigue una distribución bimodal. Los datos no parecen contener ningún valor atípico, como muestra el diagrama de caja de la figura (109).

Una vez que realizado el análisis univariante, el siguiente paso es aplicar un modelo de Regresión Lineal Múltiple, el cual se muestra a continuación.

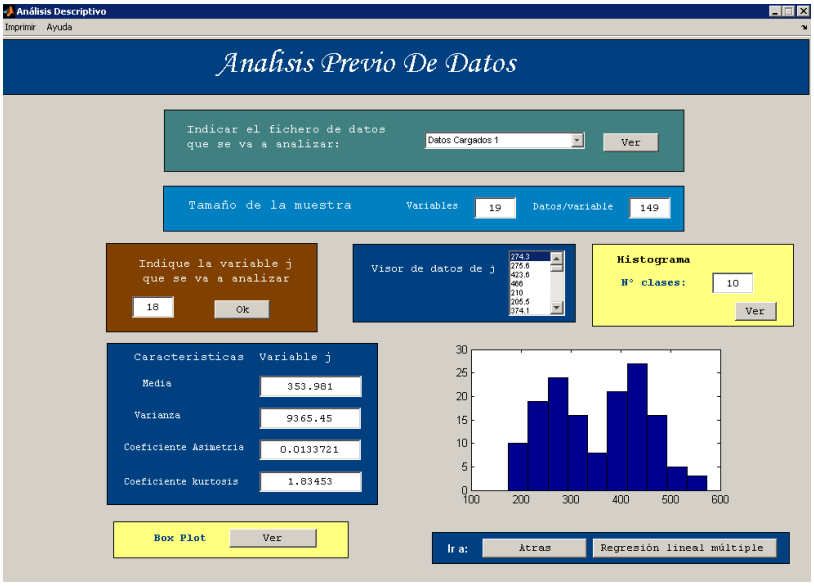


Figura 108: Análisis de la variable Fuerza de Corte (N) del proceso de taladrado.

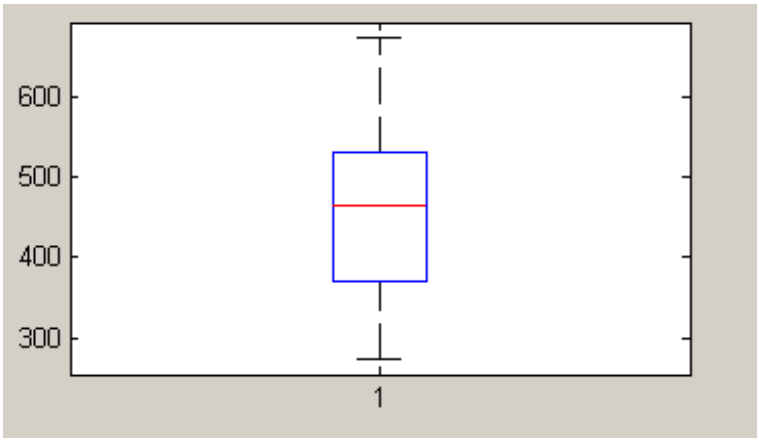


Figura 109: Diagrama de caja para los datos de Fuerza de corte (N)

5.3.4. Regresión Lineal Múltiple

La primera aplicación que permite realizar la interfaz es aplicar modelos de Regresión Lineal Múltiple. Después de aplicarlo a los datos, se comprobará su bondad de ajuste a los datos. Las variables que se van a introducir de entrada, son: ángulos de corte (en radianes), avance (mm/rev), velocidad de corte (m/min) y profundidad de pasada (mm).

Para ilustrar los resultados que proporciona la interfaz, se van a mostrar como ejemplo los datos correspondientes a la primera variable de salida, es decir, la Fuerza de corte (N). Para la otra variable de salida (Fuerza axial (N)) se realizaría el mismo proceso. Los gráficos son:

- Valores reales-Valores estimados, figura (110).
- Diagrama de dispersión Yestimado-Residuo, figura (111).
- Histograma de residuos, figura (112).

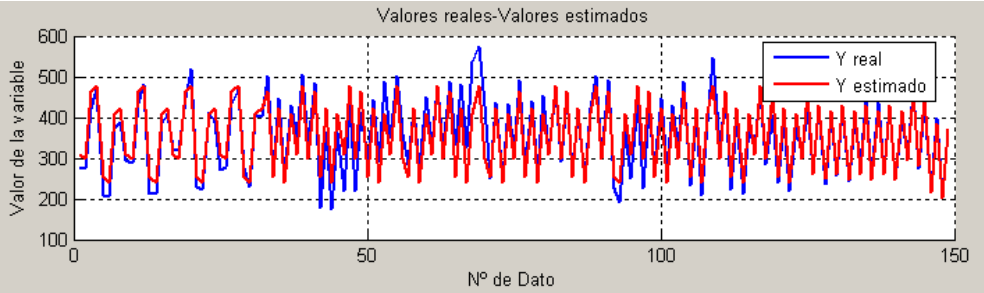


Figura 110: Valores reales - Valores estimados para la Fuerza de Corte (N) en el proceso de taladrado.

Además de los gráficos, la interfaz gráfica permite al usuario visualizar los valores de MARE % para las variables de salida, como indica la figura (113).

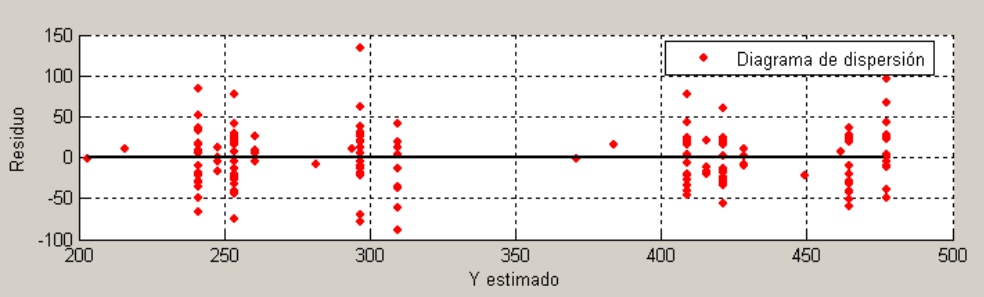


Figura 111: Residuos generados en la ejecución del modelo de Regresión Lineal Múltiple

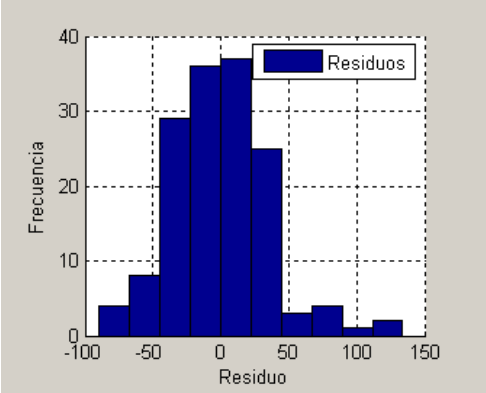


Figura 112: Histograma de residuos

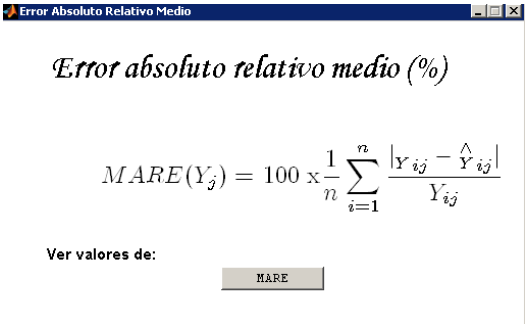


Figura 113: MARE en % para las variables de salida para el modelo de Regresión Lineal Múltiple.

De los resultados proporcionados por el modelo de Regresión Lineal Múltiple se puede decir que este modelo deja muchas dudas sobre su bondad de ajuste, pues mientras que para la variable de Fuerza de Corte (N) presenta un MARE del 22 %, en la Fuerza Axial (N) presenta un MARE de un 30 %. Estos valores de MARE % son demasiado altos, por lo tanto es necesario poner en práctica otro modelo, como puede ser el que nos presenta la interfaz MathNonParametrics, es decir, el modelo de Regresión Local Polinómica. Además, como se observa en el histograma de residuos, parece existir una cierta asimetría. Por otra parte, en la figura (111), se observa como los residuos se suelen concentrar para determinados valores de la variable estimada, es decir, no por igual para todos

5.3.5. Regresión Local Polinómica

Una vez que se ha decidido utilizar otro modelo diferente para el análisis de los datos correspondientes al proceso de taladrado, de forma análoga a el análisis realizado en la Curva de Potencia del aerogenerador, se utilizará el siguiente modelo que proporciona la interfaz MathNonParametrics, como es el modelo de Regresión Local Polinómica.

Lo primero que solicitará la interfaz MathNonParametrics para este modelo es que se le introduzcan las variables de entrada y salida, que se elegirán de forma análoga al modelo de Regresión Lineal Múltiple. Además, cuando se inserte este dato en el programa, se tomará como transformación de los datos el apartado de 'Normalizados', debido a que las variables que contienen los datos son muy diferentes entre sí.

El tamaño del conjunto de datos que servirá para validar el modelo se elegirá de forma que el número de datos no exceda de un 15 % de los datos de la muestra. Como las variables contienen 150 datos, se elegirán 25 datos para validar. El resto, o sea 125, quedarán como datos de entrenamiento para el modelo.

El siguiente paso a realizar es seleccionar los mejores valores de los parámetros del modelo: función Kernel, grado del polinomio y ancho de banda. El procedimiento a seguir es similar al del ejemplo anterior. Primero se selecciona la función Kernel, luego el grado del polinomio y por último el ancho de banda.

Los parámetros seleccionados para el modelo son **Kernel de tipo Triangular**, **grado del polinomio 5** y **ancho de banda para todas las variables de entrada igual a 0.7**. Para la selección de estos parámetros se ha usado Validación Cruzada, con 10 iteraciones.

En la figura (95) se muestran los valores de los parámetros usados al buscar el mejor ancho de banda, en un intervalo [0.5-0.8]. Igual que en el ejemplo anterior, se ha usado igual ancho de banda para todas las variables de entrada.

De manera similar al ejemplo anterior, observando el gráfico de MARE % respecto a h para los datos de validación, es posible encontrar el mejor valor de h (figura 115). En este caso, el mejor valor ha sido h=0.7, con un valor del MARE en los datos de validación igual

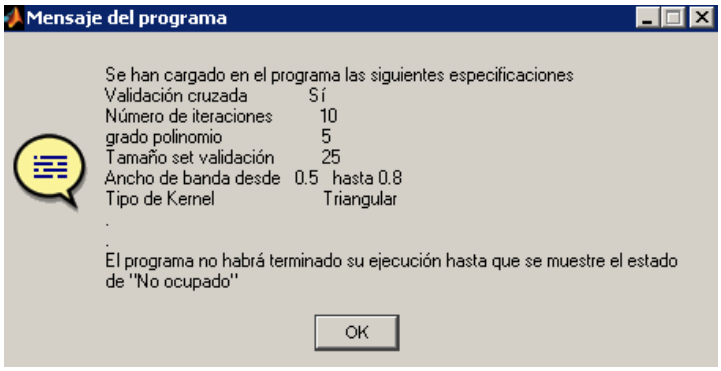


Figura 114: Parámetros utilizados para ejecutar el modelo.

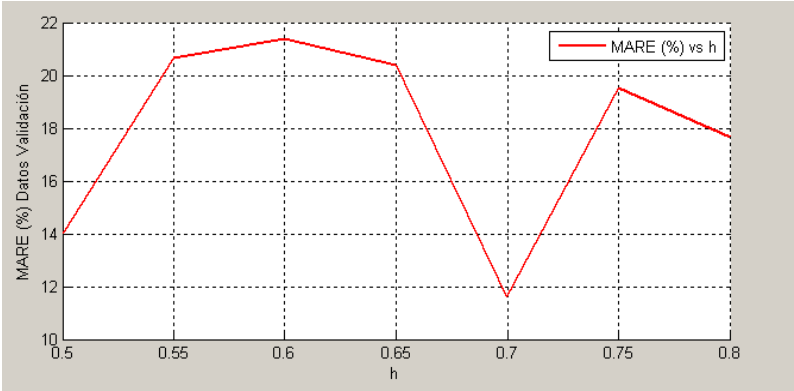


Figura 115: Evolución del MARE % respecto al ancho de banda para los datos de validación

a 12 %.

De forma análoga al análisis realizado para los resultados del modelo de Regresión Lineal Múltiple, se presentarán como ejemplo únicamente los resultados para la primera variable de salida (Fuerza de Corte (N)). Los resultados que se obtendrían para la variable de Fuerza Axial (N) se analizarían del mismo modo. Los gráficos que servirán para comprobar la bondad del modelo son los siguientes:

- Valores reales-Valores estimados, figura (116).
- Diagrama de dispersión Yestimado-Residuo, figura (117).
- Histograma de residuos, figura (118).

A la vista de estos resultados, gráficos (116), (117) y (118), se puede concluir que el modelo no es del todo adecuado. El gráfico de dispersión de los residuos presenta un comportamiento que parece tener cierta no linealidad. Además el histograma de residuos muestra una distribución asimétrica, no muy próxima a la normal. A pesar de esto, el modelo de Regresión

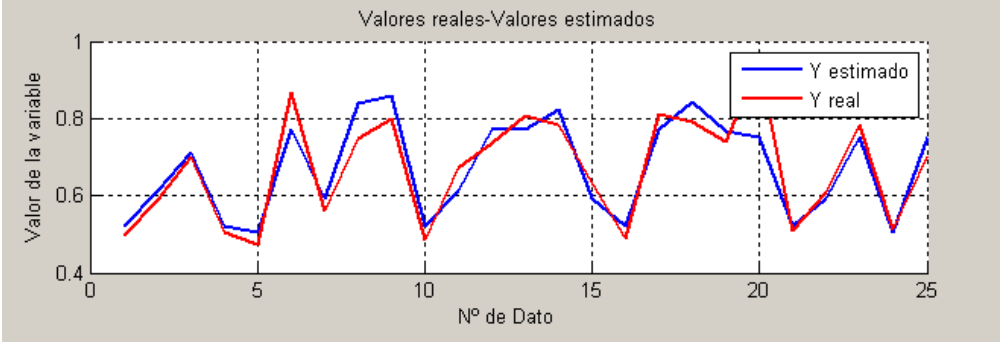


Figura 116: Gráfico Y estimado - Y real para los 25 datos de validación

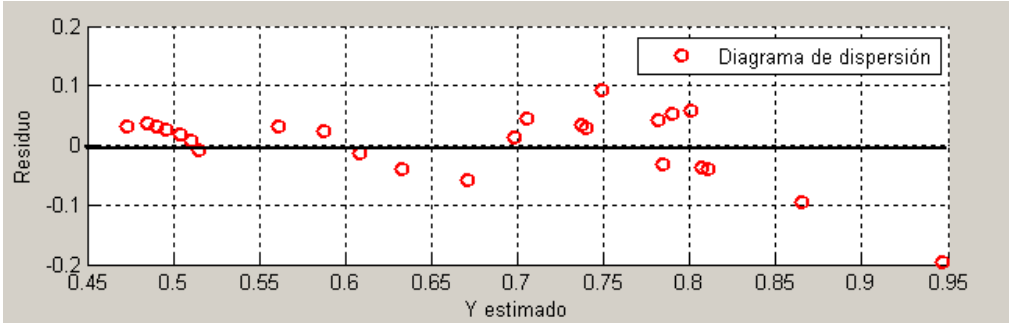


Figura 117: Gráfico de dispersión Residuos - Y estimado

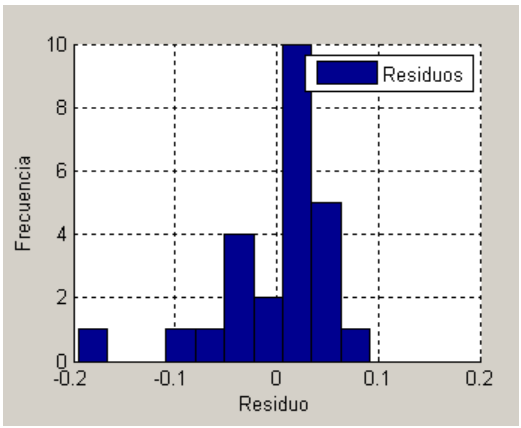


Figura 118: Histograma de residuos

Local Polinómica muestra mejores resultados (MARE en torno a un 12 %) que el modelo de Regresión Lineal Múltiple (MARE en torno a 22 %).

Con el fin de obtener un modelo más adecuado, deberían quizás incluirse nuevas variables de entrada, aumentar el número de datos, etc.

5.3.6. Estimación de nuevos datos a partir del modelo de RLPolinómica

Una vez que han sido seleccionados los parámetros del mejor modelo, en este caso:

- Tipo de Kernel: Triangular
- Grado del polinomio: 5
- Ancho de banda para las dos variables de entrada: 0.7

es posible obtener unos valores estimados de Fuerza de corte y Fuerza axial para nuevos valores de entrada.

Los valores numéricos de las entradas que van a ser utilizados con este propósito, son los mostrados a continuación:

Ángulo α (radianes)	Ángulo γ (radianes)	Velocidad corte (m/min)	Avance (mm/rev)	Profundidad pasada (mm)
0.09	1.31	240	0.05	2
0.09	1.31	240	0.1	2
0.09	1.31	120	0.1	2
0.09	1.48	240	0.1	2
0.09	1.31	120	0.05	2
0.09	1.31	120	0.1	2
0.09	1.31	240	0.05	2
0.09	1.31	240	0.1	2
0.09	1.48	120	0.05	2
0.09	1.48	120	0.1	2
0.09	1.48	240	0.05	2
0.09	1.48	240	0.1	2
0.09	1.31	120	0.05	2

Figura 119: Datos de entrada al modelo para los que se desea obtener unos datos de salida.

Igual, que en el ejemplo anterior, en este caso, se conocen los valores reales de la fuerzas para los valores de entrada de la figura (119). Estos valores reales, se muestran en la figura (120). Con ellos se calculará el error cometido con el modelo de RLP.

Los valores estimados con el modelo de RLP son los mostrados en la figura (121). El valor del error calculado como *valor estimado* − *valor* real, se muestra en la figura (122). El error absoluto relativo medio, MARE %, se muestra en la figura (123).

A la vista de los resultados, se puede concluir que:

Fuerza corte (N)	Fuerza axial (N)
380.4000	250.1000
112.6000	47.9000
111.1000	15.8000
286.1000	127.0000
307.1000	146.4000
221.8000	205.3000
218.1000	142.3000
344.9000	180.5000
420.1000	295.9000
130.1000	91.6000
122.5000	63.6000
311.7000	179.2000
288.8000	163.0000

Figura 120: Fuerzas de corte y axial reales en las condiciones de los parámetros de entrada al modelo

Fuerza corte (N)	Fuerza axial (N)
374.5290	259.3533
123.2974	41.1613
109.2367	19.2864
295.2053	125.9134
304.7869	152.6913
230.0981	201.1824
206.5109	151.4168
345.7537	179.5507
409.2294	302.3546
131.1458	83.3950
116.4124	72.7237
315.9479	174.0959
286.0235	165.9719

Figura 121: Fuerzas de corte y axial (N) estimadas.

Residuos generados:

Fuerza corte (N)	Fuerza axial (N)
5.8710	-9.2533
-10.6974	6.7387
1.8633	-3.4864
-9.1053	1.0866
2.3131	-6.2913
-8.2981	4.1176
11.5891	-9.1168
-0.8537	0.9493
10.8706	-6.4546
-1.0458	8.2050
6.0876	-9.1237
-4.2479	5.1041
2.7765	-2.9719

Figura 122: Residuos generados en la estimación del modelo

MARE %

Fuerza corte (N)	Fuerza axial (N)
1.5434	-3.6998
-9.5003	14.0682
1.6771	-22.0659
-3.1826	0.8556
0.7532	-4.2973
-3.7412	2.0056
5.3136	-6.4067
-0.2475	0.5259
2.5876	-2.1814
-0.8039	8.9574
4.9695	-14.3454
-1.3628	2.8483
0.9614	-1.8233

Figura 123: MARE % que ha proporcionado el modelo en la estimación

1. Los datos estimados están en el entorno de los datos reales, y por lo tanto el modelo se ha ejecutado de forma satisfactoria

2. En cuanto a los residuos generados, el máximo residuo en valor absoluto es de 11.58. Es un valor cercano al valor real, debido a que el orden de magnitud de las fuerzas es de centenas, y por lo tanto, la bondad de ajuste es adecuada.

3. El máximo MARE % medio para cada variable está alrededor de 10 %.

6. Conclusiones y Futuras líneas de trabajo

A continuación, se expone, de forma muy resumida un análisis sobre el cumplimiento de los objetivos propuestos y se presentan las principales conclusiones derivadas de los resultados a los que se ha llegado una vez finalizado este proyecto. También, se enumeran las posibles líneas de trabajo que puedan dar continuidad al proyecto presentado.

6.1. Conclusiones en la consecución de los objetivos propuestos

En primer lugar, el objetivo primordial era desarrollar una interfaz gráfica que permitiera al usuario estimar un modelo de RLP. La interfaz *MathNonParametrics* se ha desarrollado en base a este objetivo, de tal forma que el usuario pueda utilizarla sin muchos conocimientos previos, hasta el punto que si se utiliza de forma intuitiva, se conseguirá ponerla en funcionamiento. También, se buscaba que la interfaz pudiese ser utilizada para predecir nuevos valores a partir del mejor modelo estimado. Este objetivo se ha conseguido destinando una pantalla de la interfaz que realiza esta predicción .

En cuanto a las características de funcionamiento que se buscaban con la interfaz se puede concluir lo siguiente:

- El primer objetivo estaba relacionado con la facilidad de utilización de la interfaz. Este objetivo se ha conseguido en cada una de las pantallas de la interfaz, ya que el usuario puede ejecutar las distintas opciones de forma clara, sin necesidad de conocimientos teóricos muy profundos sobre RLP:

- El segundo objetivo fue el desarrollo de MathNonParametrics de forma secuencial, de tal forma que el usuario no se pueda perder o dudar en cuál es el procedimiento a seguir. Este objetivo se ha conseguido incluyendo en la interfaz mensajes automáticos de error o pausas que no permiten avanzar al usuario si no ha introducido toda la información necesaria en cada pantalla, o no se ha introducido de forma correcta. Así, si el usuario debido a un despiste no avanza en la dirección correcta o no introduce la información que el programa requiere de forma correcta, el programa recomendará como qué debe hacer. Además de esto, y cumpliendo con el sexto objetivo, el usuario dispone de ayuda a la que puede acceder en cualquier momento. El usuario, puede consultar información sobre por ejemplo cuál deben ser los pasos que debe seguir en cada pantalla, información teórica sobre lo que el programa está haciendo en cada momento, información sobre los gráficos obtenidos, etc.

- El tercer objetivo que era el incluir un análisis previo de los datos se ha tenido en cuenta al introducir, en primer lugar, un análisis univariante de las variables (histogramas, gráficos boxplot). Y, también, incluyendo la posibilidad de estimar un modelo

paramétrico sencillo, la Regresión Lineal Múltiple. Este análisis permitirá al usuario saber si este modelo paramétrico es adecuado a sus datos, antes de empezar con un modelo no paramétrico que resultaría menos adecuado en caso de no ser necesario.

- El cuarto objetivo fue el de conseguir que la interfaz mostrara resultados sencillos y útiles para evaluar el modelo estimado. Este objetivo se ha conseguido ya que en cada pantalla de la interfaz se ha podido maximizar la presentación de datos en forma gráfica: gráficos de dispersión, histogramas, gráficos de error medio absoluto relativo, gráficos de valor de variables estimadas respecto a reales, etc.

- El quinto objetivo que fue poder grabar e imprimir, también ha sido tenido en cuenta, ya que se permite al usuario grabar los resultados más importantes obtenidos en la estimación del modelo. Para guardar los resultados se ha destinado una pantalla especial. Para imprimir basta con pulsar un botón en la zona superior de cada pantalla.

6.2. Otras conclusiones

Tradicionalmente, la identificación de un modelo que se ajuste a unos datos es realizada con modelos de estimación paramétrica. En estos modelos, se comienza haciendo supuestos rígidos sobre la estructura básica de los datos, luego se estiman de la forma más eficiente posible los parámetros que definen la estructura y por último se comprueba si los supuestos iniciales se cumplen. Esto en numerables ocasiones puede ser útil si los datos son adecuados a las hipótesis realizadas.

La Regresión Local Polinómica, en cambio, desarrolla un 'modelo libre' para predecir la respuesta sobre el rango de valores de los datos utilizados. Básicamente está constituida por métodos que proporcionan una estimación suavizada de la relación para un conjunto de valores de las variables explicativas. Estos valores son ponderados de modo que, por ejemplo, los vecinos más cercanos tengan mayor peso que los más elejados dentro de un conjunto de datos. Se pueden utilizar diversas funciones de ponderación, que son los pesos en que se basan los estimadores. La combinación de la función de ponderación y el ancho de banda inciden en la bondad de la estimación resultante del modelo de Regresión Local Polinómica. Mediante los ejemplos que se incluyen en este trabajo, se ha podido comprobar que en problemas como éstos, la RLP resulta ser una herramienta más adecuada que la regresión lineal múltiple (modelo paramétrico), por ejemplo.

Por otra parte, otra conclusión que es importante destacar, es que la interfaz MathNonParametrics puede ser utilizada para la resolución de problemas de distintos sectores, tales como la medicina, ingeniería, economías, etc; sectores en los que quizás los usuarios tienen poco conocimiento de estas técnicas estadísticas, de manera que el uso de la interfaz les facilitará mucho su aplicación.

6.3. Futuras líneas de trabajo

Como futuras líneas de trabajo se han pensado varias posibilidades, encaminadas a ampliar el software que proporciona este proyecto y que sirva de base a futuras ampliaciones. Algunas de estas ampliaciones pueden ser las siguientes:

1. Futuras modificaciones para *MathNonParametrics*:

   Uno de los aspectos más interesantes que se podrían incorporar sería la adquisición de datos reales en tiempo real, con la ayuda de algún periférico de recogida de datos y mediante la herramienta Data Tool Box de MatLab.

   Otra mejora posible es convertir la interfaz *MathNonParametrics* en un programa ejecutable sin necesidad de tener instalado el programa MatLab en el ordenador, es decir, convertir la interfaz en un programa independiente de Matlab.

2. Implementar otros modelos no paramétricos (splines, wavelets, etc) de manera que se puedan comparar los resultados de una técnica con respecto a otra.

3. Incorporar un algoritmo que permita desarrollar la puesta a punto del modelo de Regresión Local Polinómica sin necesidad que el usuario cambie parámetros para mejorar la respuesta del modelo. Esto mejoraría el tiempo necesario para ejecutar el modelo, debido a que el ordenador buscaría los parámetros óptimos.

Referencias

[1] Peña, Daniel (2008). Fundamentos de Estadística. Alianza Editorial

[2] Härdle, W. (1991). Smoothing Techniques, With Implementations in S, Springer, New York.

[3] Härdle W., Müller M., Sperlich S. y Werwatz A. (2004). Nonparametric and Semiparametric Models An Introduction, Springer, New York.

[4] Nadaraya, E. A. (1964). On estimating regression, *Theory of Probability and its Applications* 10: 186-190.

[5] Park, B. U. & Turlach, B. A. (1992) Practical performance of several data driven bandwidth selectors, *Computational Statistics* 7: 251-270.

[6] Watson, G. S. (1964). Smooth regression analysis, Sankhy, *Series A* 26: 359-372.

[7] Smith, Scott T. (2006). Matlab advanced gui development .Dog Ear Publishing.

[8] Javier García Galón. (2007) Aprenda MatLab 7.0 como si estuvieras en primero. Universidad Politécnica Madrid Ediciones